

ファジィ集合による階層的適応度を応用した動的分散遺伝的アルゴリズムを用いた軌道生成

著者	市毛 優智
発行年	2019-01
その他のタイトル	Research of Guidance Trajectory Generation using Dynamically Distributed Genetic Algorithm which Applied Hierarchical Fitness with the Fuzzy Method
学位授与年度	平成30年度
学位授与番号	17104甲工第464号
URL	http://hdl.handle.net/10228/00007667

博士学位論文

ファジィ集合による階層的適応度を応用した
動的分散遺伝的アルゴリズムを用いた軌道生成

平成 31 年 1 月

工学府 工学専攻
機械知能工学領域 15595104

市毛 優智

目次

目次	i
図番号一覧.....	iv
表番号一覧.....	vii
第 1 章 緒論	1
1.1. 本研究の背景.....	1
1.2. 本研究の目的.....	3
1.3. 本研究の概要.....	5
1.4. 本論文の構成.....	6
第 2 章 GA を用いた最適軌道生成	7
2.1. はじめに.....	7
2.2. 最適化問題の設定.....	8
2.2.1. 軌道生成問題の設定	8
2.2.2. 有翼ロケットの運動方程式.....	9
2.3. 動的分散遺伝的アルゴリズム	12
2.3.1. 遺伝的アルゴリズム	12
2.3.2. 分散遺伝的アルゴリズム	16
2.3.3. 動的分散遺伝的アルゴリズム	18
2.3.4. 遺伝子の設定	19
2.3.5. 適応度	22
2.3.6. 特徴量	23
2.3.7. 個体間の距離の算出と非類似度.....	24
2.3.8. 階層的クラスタリング	25
2.3.9. DynDGA を用いた誘導軌道生成	27

2.3.10. おわりに.....	32
第3章 ファジィ集合による階層的適応度を応用した動的分散遺伝的アルゴリズムを用いた軌道生成と係数自動調整手法の提案.....	33
3.1. はじめに.....	33
3.2. ファジィ集合を用いた適応度計算.....	35
3.2.1. ファジィ集合.....	35
3.2.2. 軌道生成におけるファジィ集合の適用.....	36
3.2.3. ファジィ集合を用いた適応度.....	40
3.2.4. ファジィ集合を用いた適応度による軌道生成シミュレーション.....	40
3.3. 階層的手法を用いた適応度計算.....	47
3.3.1. 評価指標の優先度と階層的手法.....	47
3.3.2. 先制優先因子.....	47
3.3.3. ファジィ集合を用いた階層的適応度計算手法による軌道生成シミュレーション 48	
3.4. 適応度に使用する係数の自動調整手法.....	53
3.5. おわりに.....	59
第4章 DynDGA を用いたリアルタイム最適軌道生成システム実用化の検討.....	60
4.1. はじめに.....	60
4.2. FPGA を用いた GA によるオンボードリアルタイム軌道計算.....	61
4.2.1. FPGA.....	61
4.2.2. 実飛行環境下における軌道生成実証飛行.....	61
4.2.3. カイトプレーンに搭載する試験装置.....	63
4.2.4. 飛行試験場所.....	66
4.2.5. GA 計算条件.....	67
4.2.6. 飛行試験結果.....	70
4.3. DynDGA を用いた誘導軌道再生シミュレーション.....	72

4.3.1.	誘導軌道再生成シミュレーション	72
4.3.2.	シミュレーションの構成	72
4.3.3.	機体モデル	74
4.3.4.	飛行領域.....	76
4.3.5.	シミュレーション結果.....	77
4.4.	DynDGA を用いたリアルタイム誘導軌道生成システムの検討	82
4.4.1.	実飛行を考慮したリアルタイム化の検討.....	82
4.4.2.	リアルタイム性を考慮した誘導軌道再生成シミュレーションの条件	85
4.4.3.	リアルタイム性を考慮した誘導軌道再生成シミュレーション結果.....	87
4.5.	おわりに.....	90
第 5 章	結論	92
5.1.	本研究でわかったこと.....	92
5.2.	今後の課題	93
謝辞	94
参考文献	95

図番号一覧

図 2.2-1	有翼ロケットの飛行フェーズ	8
図 2.2-2	軌道生成問題の設定	9
図 2.2-3	有翼ロケット	9
図 2.2-4	直行座標系の定義	10
図 2.2-5	有翼ロケットの空力特性	11
図 2.3-1	遺伝的アルゴリズムの処理の流れ	13
図 2.3-2	世代交代モデル	14
図 2.3-3	エリート選択 (ER)	14
図 2.3-4	ブレンド交叉	15
図 2.3-5	一様突然変異	16
図 2.3-6	DGA アルゴリズム	17
図 2.3-7	DGA の移住モデル	17
図 2.3-8	DynDGA の処理イメージ	18
図 2.3-9	フーリエ級数の数値的制限	20
図 2.3-10	正規化したフーリエ級数の例	21
図 2.3-11	最適解が境界条件付近の場合の例	21
図 2.3-12	軌道最適化に特化したフーリエ級数の制御入力表現	22
図 2.3-13	適応度評価項目イメージ	22
図 2.3-14	ダウンレンジとクロスレンジを時間の関数とする特徴量の定義	24
図 2.3-15	ユークリッド (平方) 距離の例	24
図 2.3-16	階層的クラスタリングにおけるデンドログラム	26
図 2.3-17	Ward's Method	26
図 2.3-18	シミュレーションの飛行領域	28
図 2.3-19	DynDGA を用いた誘導軌道最適化結果(最適軌道)	29
図 2.3-20	DynDGA を用いた誘導軌道最適化結果(高度プロファイル)	29
図 2.3-21	DynDGA を用いた誘導軌道最適化結果(速度プロファイル)	30
図 2.3-22	DynDGA を用いた誘導軌道最適化結果(迎角プロファイル)	30
図 2.3-23	DynDGA を用いた誘導軌道最適化結果(バンク角プロファイル)	31
図 2.3-24	DynDGA を用いた誘導軌道最適化結果(平均適応度)	31
図 2.3-25	DynDGA を用いた誘導軌道最適化結果(最終世代の軌道)	32
図 3.2-1	直角台形型メンバーシップ関数	35
図 3.2-2	到達地点誤差のファジィ化イメージ	37

図 3.2-3	到達地点方位角誤差メンバシップ関数イメージ	38
図 3.2-4	飛行禁止区域のメンバシップ関数	39
図 3.2-5	凸台形型メンバシップ関数	40
図 3.2-6	ファジィ集合を用いた誘導軌道最適化結果(最適軌道)	42
図 3.2-7	ファジィ集合を用いた誘導軌道最適化結果(高度プロファイル)	42
図 3.2-8	ファジィ集合を用いた誘導軌道最適化結果(速度プロファイル)	43
図 3.2-9	ファジィ集合を用いた誘導軌道最適化結果(迎角プロファイル)	43
図 3.2-10	ファジィ集合を用いた誘導軌道最適化結果(バンク角プロファイル)	44
図 3.2-11	ファジィ集合を用いた誘導軌道最適化結果(平均適応度)	44
図 3.2-12	ファジィ集合を用いた誘導軌道最適化結果(メンバシップ関数値)	45
図 3.2-13	ファジィ集合を用いた誘導軌道最適化結果(最終世代の軌道)	45
図 3.2-14	優先順位が反転するパターン	46
図 3.3-1	階層的手法による軌道最適化結果(最適軌道)	48
図 3.3-2	階層的手法による軌道最適化結果(高度プロファイル)	49
図 3.3-3	階層的手法による軌道最適化結果(速度プロファイル)	49
図 3.3-4	階層的手法による軌道最適化結果(迎角プロファイル)	50
図 3.3-5	階層的手法による軌道最適化結果(バンク角プロファイル)	50
図 3.3-6	階層的手法による軌道最適化結果(平均適応度)	51
図 3.3-7	階層的手法による軌道最適化結果(メンバシップ関数値)	51
図 3.3-8	階層的手法による軌道最適化結果(最終世代の軌道)	52
図 3.4-1	係数自動調整手法による軌道最適化結果(最適軌道)	54
図 3.4-2	係数自動調整手法による軌道最適化結果(高度プロファイル) ...	55
図 3.4-3	係数自動調整手法による軌道最適化結果(速度プロファイル) ...	55
図 3.4-4	係数自動調整手法による軌道最適化結果(迎角プロファイル) ...	56
図 3.4-5	係数自動調整手法による軌道最適化結果(バンク角プロファイル)	56
図 3.4-6	係数自動調整手法による軌道最適化結果(平均適応度)	57
図 3.4-7	係数自動調整手法による軌道最適化結果(メンバシップ関数値)	57
図 3.4-8	係数自動調整手法による軌道最適化結果(最終世代の軌道)	58
図 4.2-1	Internal Constitution of FPGA	61
図 4.2-2	カイトプレーン	62
図 4.2-3	小型有翼ロケット実験機 WIRES#014	62
図 4.2-4	カイトプレーン飛行イメージ	64
図 4.2-5	プロトタイプデバイスシステム図	65

図 4.2-6	誘導システム内部の構成.....	65
図 4.2-7	飛行試験実施場所外観.....	67
図 4.2-8	定常滑空近似概要.....	68
図 4.2-9	定常滑空近似から得られる軌道例.....	69
図 4.2-10	FPGA 外観.....	69
図 4.2-11	カイトプレーン飛行試験時に生成された誘導軌道（平面図）	70
図 4.2-12	カイトプレーン飛行試験時に生成された誘導軌道（鳥瞰図）	71
図 4.3-1	誘導軌道再生成シミュレーション概要.....	72
図 4.3-2	誘導軌道再生成シミュレーションのフェーズ.....	73
図 4.3-3	各フェーズにおける誘導コマンド.....	73
図 4.3-4	滑空帰還フェーズ中の誘導システム処理概要.....	74
図 4.3-5	誘導軌道再生成シミュレーションの機体モデル.....	75
図 4.3-6	HIMES 空力パッケージ Ver. 1.4 基本空力係数.....	76
図 4.3-7	誘導軌道再生成シミュレーションの飛行領域.....	77
図 4.3-8	誘導軌道再生成シミュレーション結果 A.....	78
図 4.3-9	誘導軌道再生成シミュレーション結果 B.....	79
図 4.3-10	誘導軌道再生成の様子.....	80
図 4.3-11	迎角コマンド(赤)と飛行結果(青).....	80
図 4.3-12	バンク角コマンド(赤)と飛行結果(青).....	81
図 4.3-13	スピードブレーキコマンド(赤)と飛行結果(青).....	81
図 4.4-1	リアルタイム性を考慮した誘導軌道生成システム概要.....	84
図 4.4-2	WIRES#015 モデル.....	85
図 4.4-3	誘導軌道再生成シミュレーション飛行環境.....	86
図 4.4-4	再生成シミュレーション結果(飛行軌道).....	88
図 4.4-5	再生成シミュレーション結果(高度プロファイル).....	88
図 4.4-6	再生成シミュレーション結果(速度プロファイル).....	89
図 4.4-7	再生成シミュレーション結果(制御入力プロファイル).....	89
図 4.4-8	再生成シミュレーション結果(メンバシップ関数値).....	90

表番号一覧

表 2.2-1	有翼ロケットの機体諸元.....	9
表 2.3-1	DynDGA 軌道生成シミュレーション条件	28
表 3.2-1	メンバシップ関数の設定.....	41
表 3.2-2	ファジィ集合を用いた手法のシミュレーションにおける重み係 数.....	41
表 4.2-1	カイトプレーン機体諸元.....	62
表 4.2-2	小型有翼ロケット実験機 WIRES#014 機体諸元	63
表 4.2-3	誘導システムへ入力するデータ	66
表 4.2-4	誘導システムから出力されるデータ	66
表 4.2-5	カイトプレーン飛行環境とシミュレーション使用値.....	67
表 4.2-6	カイトプレーン飛行試験の GA 計算条件	68
表 4.2-7	FPGA 諸元.....	69
表 4.2-8	カイトプレーンによる飛行実証結果.....	70
表 4.3-1	誘導軌道再生成シミュレーションの機体諸元.....	75
表 4.3-2	各舵角範囲	76
表 4.4-1	計算条件比較.....	82
表 4.4-2	FPGA 並列化における機能使用率.....	83
表 4.4-3	WIRES#015 機体諸元	86
表 4.4-4	誘導軌道再生成シミュレーションの評価項目優先度設定	87

第1章 緒論

1.1. 本研究の背景

人類がはじめて宇宙空間に人工物を到達させたのは 1942 年のことだった。ドイツの技術者、フォン・ブラウンらは 1927 年に結成されたドイツ宇宙旅行協会 VfR (Verein für Raumschiffahrt) において宇宙旅行を目指す液体燃料ロケットの研究を行い、その 15 年後に A4 ロケットの打ち上げに成功させたのだ。また、電子回路を装備し、ジャイロスコープによる慣性飛行を成功させ、これまでとは一線を画した精度と成功率によって世界を震撼させた[1]。以降、第二次世界大戦で世界は混乱の渦へと陥るが、弾道ミサイルとしてロケットは研究開発が行われてきた。

終戦後、ソビエト連邦とアメリカ合衆国の宇宙開発競争が激化し、ソビエト連邦の人類初人工衛星「スプートニク 1 号」やアメリカ航空宇宙局 NASA (National Aeronautics and Space Administration) のアポロ計画による有人月面着陸が行われ、世界の宇宙輸送技術は日進月歩であった。特にアポロ計画では、コンピュータによる自動制御を世界で初めて取り入れ、IC 技術の発展に大きく貢献することとなり、宇宙だけでなく民間技術の発展の礎となった。NASA は 1960 年代末頃には、サターン V 型ロケットの技術を応用した再使用型の宇宙往還機の開発を考え始めており、これがスペースシャトルの基本的構想となった。同時に、宇宙往還機の開発は世界各国でも進められ、ソ連の無人宇宙往還機「ブラン」、欧州宇宙機関 ESA (European Space Agency) の有人有翼回収機「エルメス」等研究が行われた[2]。

近年では、民間企業による再使用型宇宙往還機の開発が活発であり、The Spaceship Company (TSC) の SpaceShip Two やスペース X (Space Exploration Technologies Corp.) の Falcon 9 が実飛行を成功させている。特に Falcon 9 は、既に ISS との連絡に成功し、1 段目の再使用を何度も行っている。

日本も、1980 年代から日本独自の再使用型宇宙輸送システムの開発を目指して、宇宙航空研究開発機構 JAXA (Japan Aerospace eXploration Agency) の前身である宇宙開発事業団 NASDA (National Space Development Agency of Japan)、宇宙科学研究所 ISAS (Institute of Space and Astronautical Science)、航空宇宙技術研究所 NAL (National Aerospace Laboratory of Japan) が各々研究を行ってきた。

NASDA 及び NAL は、無人の再使用型有翼回収機「HOPE (H-II Orbiting PlanE)」, その後に宇宙往還技術試験機「HOPE-X (H-II Orbiting PlanE eXperiment)」の開発を進めた。HOPE の特徴は、姿勢制御用と軌道離脱用の小型スラスタを備えては

いるが、推進用のメインエンジンは持っておらず、軌道投入自体は H-II ロケットにて行われる点にある。軌道上でのミッションを終了後、軌道離脱、大気圏再突入、滑空帰還、水平着陸を行う計画であった。また、軌道突入フェーズから超音速飛行、滑空誘導飛行と幅広い飛行領域を持つ機体を一度に開発するのは難しいため、技術実証機として、大気圏再突入時の空力加熱防護技術を実証する OREX (Orbital Reentry EXperiment), Lifting Body の極超音速域での誘導制御技術等の飛行実証を目的とした HYFLEX (HYpersonic FLight EXperiment), 自動着陸技術の実証を目的とした ALFLEX (Automatic Landing FLight EXperiment), 最後には遷音速での空力特性と姿勢制御技術の実証を目的とする HSFD (High Speed Flight Demonstration) が実施された。

ISAS では、無人で弾道飛行を行う単段式の有翼飛翔体であり、二段式の宇宙往復輸送システムの一段目に当たるフライバックブースターに発展可能な有翼式再使用型観測ロケット HIMES (Highly Maneuverable Experimental Space Vehicle) の研究を行っていた。HIMES は空間に対して静止可能で中・高層大気の観測等を行った後、大気圏に再突入し発射点近傍の滑走路に滑空帰還を行って水平に着陸する。また、弾道軌道から大気圏に再突入するため、大気圏再突入速度が小さく、本格的な耐熱システムは必要としない設計だった。

各々の機関で行われていた研究は、概念設計を踏まえて各種飛行実験を実施するに至るまで進んだものの、実機の開発を待たずに計画は中止されてしまった。しかし、開発のために行われたさまざまな実験により蓄積されたデータや各種技術は、将来の宇宙往還機の開発に有用である。

現在、JAXA の宇宙科学研究所が、鈍頭形状の垂直離着陸式再使用型観測ロケットのスケール機を用いた飛行実験を行っている。しかし、垂直離着陸式よりも、翼による揚力を利用する有翼式の方が、帰還時にエンジンの逆噴射が必要なく、フライバックして任意の着陸地点まで滑空飛行する帰還能力に優れており、運用コストの低減等の経済的な効果についても魅力がある。また、宇宙空間への物資輸送や化学実験、刊行だけでなく大陸間弾道飛行などへの転用も可能であることから、航空宇宙市場の拡大に貢献することが期待できる。

1.2. 本研究の目的

有翼式宇宙往還機の実現を見据えると、アボート飛行を含む自在な飛行運用のためにも、リアルタイムに最適軌道生成が可能な最適誘導システムは重要研究課題の一つである。極超音速での再突入から亜音速の進入着陸に至る帰還軌道生成手法には多くの従来研究があるが、その中でも基準軌道を予め設定し、飛行中に軌道の微少な変動分に対して最適化する手法が一般的である[3, 4]。しかしミッションを中断する場合などを含めた、状況に応じた基準軌道の事前設定は困難である。このようなことから、本研究は、風等の飛行環境による飛行分散のみならず、アボート飛行のような不測の事態においても安全に機体を目的地まで誘導させるために必要な技術の研究開発を目的として、基準軌道を必要とせず飛行環境に適合した軌道をリアルタイムかつオンボードで生成可能な誘導システムの開発を目的とする。

近年研究されている逐次二次計画法等の勾配法を用いた数値解析的な手法は、計算機能力の向上により、運動方程式の微分式をそのまま積分することによって高精度かつ高速に飛行軌道の生成が行えるようになったものの、多峰性を有するような問題に対して局所解に陥りやすいこと、数値的な発散が生じる可能性があること等の課題を克服することが難しい。その一方で、これまでの先行研究により、有翼ロケットの滑空飛行フェーズを対象に、遺伝的アルゴリズム GA (Genetic Algorithm) を試みたところ、柔軟かつ大域的な探索解が得られることが分かり、また飛行禁止区域が存在する飛行環境条件においても誘導軌道生成が可能であることを示すことが出来た[5-8]。その成果として、進化計算の世代交代のアルゴリズムを変えると、得られる最適軌道が異なることがわかってきた。例えば、エリート交叉アルゴリズムでは小回りの軌道が生成されやすく、他のアルゴリズムでは大回りの軌道になりやすい等である。このような世代交代アルゴリズムに依存するような最適解の出方の差異は、好ましいものではない。世代交代の設計法の解への影響は本来起こるべきでなく、個体集団の多様性を維持しつつ収束させる必要がある。

このような問題に対して、多様性の維持と収束性を両立しつつ、効率的に最適化を行うことを目的とした動的分散遺伝的アルゴリズム DynDGA (Dynamically Distributed Genetic Algorithm) が発案された[9]。これは、母集団をサブ母集団(島)に分割し、各島で個別に世代交代を行うものである。ベースとなった分散遺伝的アルゴリズム DGA (Distributed Genetic Algorithm) では、島の数を事前に設定するのが一般的であったが[10-14]、優良でないサブ母集団を維持し続けなければならない点や逆に特徴の違う解を同一サブ母集団の中で世代交代しなければならない点などの問題点があった。そこで DynDGA では軌道の特徴より非類似度を

算出し、この非類似度から母集団からサブ母集団へと分割することにより、島の数が動的に変化する効率的な最適化が行えるようになった[9].

この DynDGA を誘導システムへ用いることにより、複数種類の最適誘導軌道を同時生成することが可能となった.

しかし、DynDGA は計算コストが非常に嵩み、一般の GA で利用される適応度の重み係数や突然変異の確率などに加え、クラスタリングの閾値や移住間隔などの設定値が必要であるため、リアルタイム性と汎用性の面を克服するための重要な課題となっている. 特に飛行環境など最適化問題の設定が変化するたびに調整が必要となる適応度の重み係数やクラスタリングの閾値などの設定値数の増加は、どのような飛行環境であっても誘導軌道の生成がリアルタイムに可能でなければならない本誘導システムの実現において大きな障害である.

このような研究課題を背景に、本研究は、ファジィ集合による階層的適応度を応用した設定値の自動調整手法の提案とオンボードでリアルタイム計算が実行可能な実用化への課題解決が本研究の最終目標である.

1.3. 本研究の概要

本研究は、繰り返しシミュレーションを試行していく中で手動設定する必要のあるクラスタリングの閾値に代表される設定値を極限まで削減し、飛行環境の変化により変更する必要のある適応度の重み係数などの設定値についても自動調整する手法として、ファジィ集合による階層的適応度を応用した設定値の自動調整手法を提案するものである。この提案手法では、ユーザーは生成された軌道の優劣を評価する項目の優先度を事前に設定するだけで、最適化の中で解の各評価項目値によって設定値の調整が自動で行われながら最適解が探索される。すなわち、これまで必要とされていた設定値を手動で繰り返し模索する必要がなくなったことから、ファジィ集合による階層的適応度を応用することで、DynDGA の汎用性が一気に向上したと言える。

リアルタイム性を向上する課題については、適応度および特徴量を計算するために、それぞれの個体についての運動方程式を積分することによって飛行軌道を算出する必要があることから、計算コストを原理的に改善することは難しいことが分かっている。すなわち、飛行距離が長くなるほど軌道計算に必要な積分ステップ数が増加するために、計算コストがより嵩んでしまうことは避けられない。ファジィ集合による階層的適応度を応用した DynDGA においても、誘導軌道の初期位置によって最適計算に必要な計算コストが運動方程式の積分ステップ数に大きく左右されてしまうことから、通常の GA で採用されているように進化が十分進んだとする世代数を最適化終了条件とするのは、実用的ではないことが分かってきた。そこで、飛行中に最適化計算を実行し続け、誘導軌道更新周期でその時点までに得られた最適解を引き継ぎ、再利用することを前提として、最適化計算終了条件を計算時間へと変更することを提案するに至った。このような最適解の継続的な引き継ぎによって計算コストの低減を図りリアルタイム性を確保する方法においても、評価項目を精度よく満足させながら、目標地点へ到達可能とする誘導軌道の生成が可能であることを飛行シミュレーション上で実証することができた。

1.4. 本論文の構成

本論文の構成を以下に述べる．

第 2 章では，先行研究の動的分散遺伝的アルゴリズム DynDGA を用いた最適軌道生成手法と軌道生成例について，その特徴をまとめた．

第 3 章では，ファジィ集合による階層的適応度を応用した係数自動調整手法を提案し，誘導軌道生成手法に適用した場合の例と共にその有用性を示した．

第 4 章では，実際に有翼ロケットへ本誘導軌道生成システムを組み込むにあたり，リアルタイム性をどのように確保するかについて検討し，ソフトウェアシミュレーションの結果を示した．

第 5 章では，新規性および独自性の観点から，研究成果をまとめた．

第2章 GA を用いた最適軌道生成

2.1. はじめに

第1章でも述べたように、従来の宇宙輸送システムの軌道計算は、基準軌道を運動方程式の近似解析解より予め設定し、飛行中に現在の飛行軌道と基準軌道との微少な変動分について最適化する手法が一般的であるものの、アボート飛行のような想定外の状況において予め基準軌道を設定しておくことは困難である。しかし、将来的に有人での飛行も視野にある再使用型有翼ロケットにおいては、このような状況においても安全な帰還飛行を自立飛行にて確実に行える必要があるため、基準軌道を用いない手法が好ましい。従来研究より、勾配法の一つである逐次二次計画法 SQP (Sequential Quadratic Programming method) を用いた誘導軌道生成手法では、単純な飛行条件において非常に高速かつ高精度な最適軌道の算出が可能であることが示されたが、飛行禁止区域等多峰性に富んだ最適化問題においては、数値解析に解いていることから、解が発散する危険性をもつことが解った。

このようなことから、本論文では大域的かつ柔軟な探索が可能な遺伝的アルゴリズム GA (Genetic Algorithm) を用いた最適軌道生成手法を用いている。GA は生命の進化を工学的にモデル化したものであり、勾配法を代表とする数値解析的な手法に比べ、解が発散しにくい特徴を持つものの、世代交代の中で生成された解集団について、個々に適応度を計算する必要があるため、計算量が膨大になってしまう欠点を持つ。GA を最適軌道生成に適用するに当たり、世代交代モデルや遺伝子の設定、突然変異などの設定値には多くの種類が存在するが、従来研究より、誘導軌道生成に最適な手法を採用している。また、一般的な GA の手法では発生しがちであった局所解への収束に対して、母集団を動的に分割統合する遺伝的アルゴリズム DynDGA (Dynamically Distributed Genetic Algorithm) が発案され、解の多様性と解の収束性を両立しながら効率よく複数の最適解軌道を生成することが可能となった。一方で、GA を用いた誘導システムをオンボードシステムに実装し、実飛行の中で誘導軌道の生成が行えるかの実証実験も行い、その有用性が確認された。

本章では、最適化問題の設定について示すと共に、従来研究の手法と結果について述べる。

2.2. 最適化問題の設定

2.2.1. 軌道生成問題の設定

有翼ロケットの最適化は図 2.2-1 に示すように揚力を用いて滑空飛行により目的地へ飛行する帰還フェーズを対象とする。

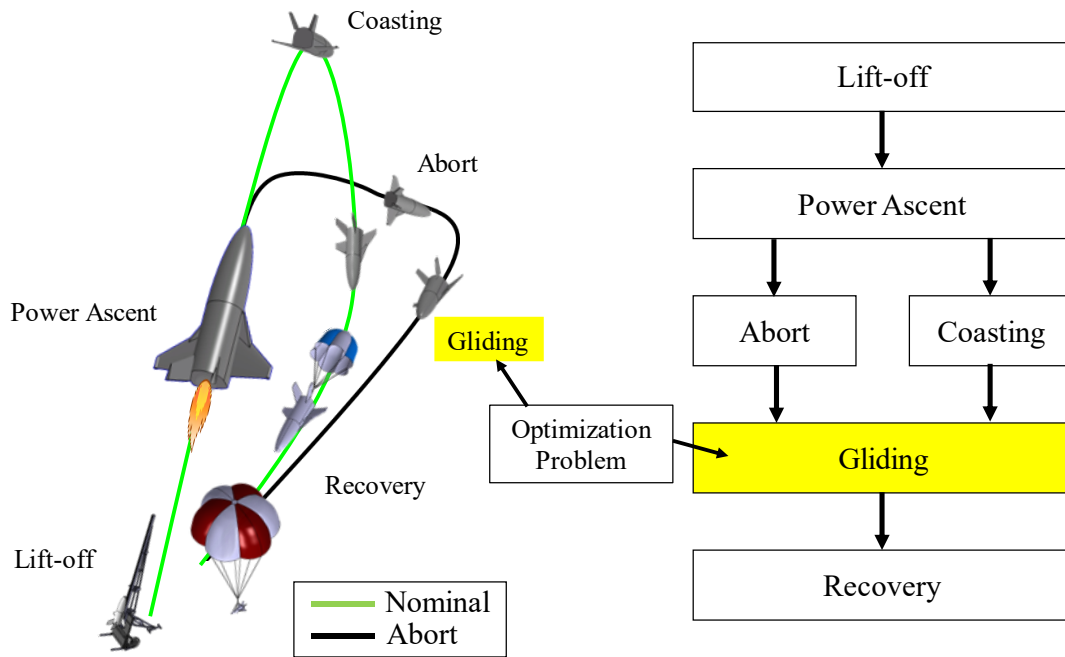


図 2.2-1 有翼ロケットの飛行フェーズ

この滑空帰還フェーズに対して図 2.2-2 のような軌道生成問題を設定する。図 2.2-3 および表 2.2-1 を代表値とするような有翼ロケットが推力なしで目標地点へ滑空飛行し帰還する軌道を生成する問題とする。さらに、飛行領域に内部の飛行を禁止する飛行禁止区域の設定をする。最適化計算は飛行禁止区域を回避し、かつ飛行時間および終端位置と終端方位角の目標値との誤差を最小化するように解の探索を行う。

なお、本論文では最適化問題の一例のみを示しているが、実際には様々な機体諸元および飛行領域において本アルゴリズムが有効であるかの確認を行っている。

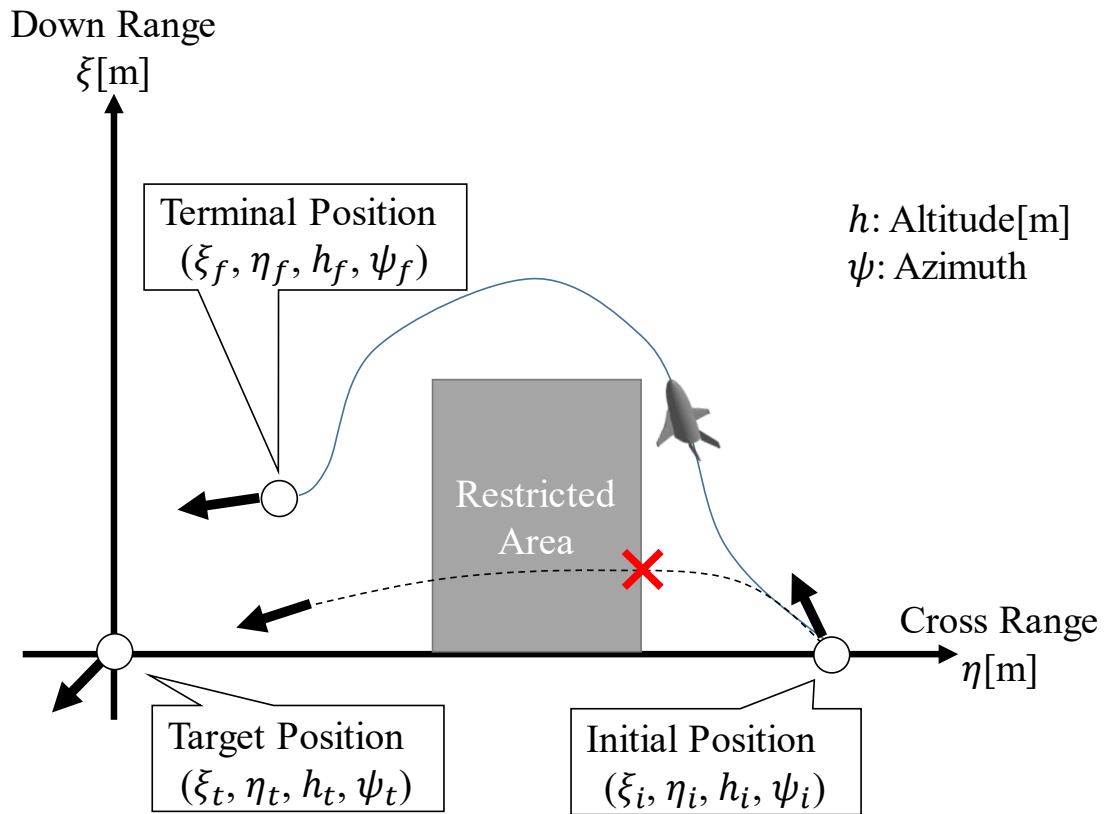


図 2.2-2 軌道生成問題の設定

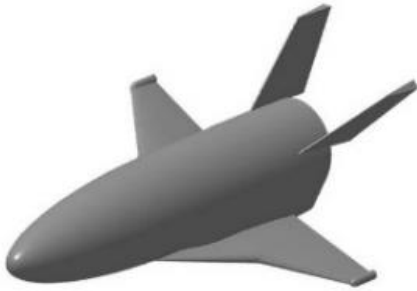


図 2.2-3 有翼ロケット

表 2.2-1 有翼ロケットの機体諸元

m	Body Mass	1000	[kg]
S_w	Wing Area	2.68	[m ²]

2.2.2.有翼ロケットの運動方程式

解探索中に使用するロケットの運動方程式を式(2.2-1)-式(2.2-6)に示す. [15]

$$m \frac{dV}{dt} = -\frac{1}{2} \rho V^2 S_w C_D - mg \sin(\gamma) \quad (2.2-1)$$

$$mV \frac{dy}{dt} = \frac{1}{2} \rho V^2 S_w C_L \cos(\phi) - mg \cos(\gamma) \quad (2.2-2)$$

$$mV \cos(\gamma) \frac{d\psi}{dt} = -\frac{1}{2} \rho V^2 S_w C_L \sin(\phi) \quad (2.2-3)$$

$$\frac{dh}{dt} = -V \sin(\gamma) \quad (2.2-4)$$

$$\frac{d\xi}{dt} = V \cos(\gamma) \cos(\psi) \quad (2.2-5)$$

$$\frac{d\eta}{dt} = V \cos(\gamma) \sin(\psi) \quad (2.2-6)$$

有翼ロケットの運動は質点系とし，図 2.2-4 に示す直交座標系で扱う．また，風の影響は誘導システム内では考慮しないものとする．地表は平面，重力 g は一定とし，空気密度 ρ は高度のみに依存した式(2.2-7)に示す様な関数で定義する．

$$\begin{cases} h \leq 20000 & \rho = ah^2 + bh + c \\ 20000 < h \leq 50000 & \rho = ah^6 + bh^5 + ch^4 + dh^3 + eh^2 + fh + g \\ 50000 < h & \rho = ah^6 + bh^5 + ch^4 + dh^3 + eh^2 + fh + g \end{cases} \quad (2.2-7)$$

t	: 時間	[s]	γ	: 経路角	[rad]
C_D	: 抗力係数	[-]	ψ	: 方位角	[rad]
C_L	: 揚力係数	[-]	ξ	: ダウンレンジ	[m]
g	: 重力加速度	[m/s ²]	η	: クロスレンジ	[m]
m	: 質量	[kg]	h	: 高度	[m]
S_w	: 主翼基準面積	[m ²]	α	: 迎角	[rad]
ρ	: 大気密度	[kg/m ³]	ϕ	: バンク角	[rad]
V	: 機体速度	[m/s]			

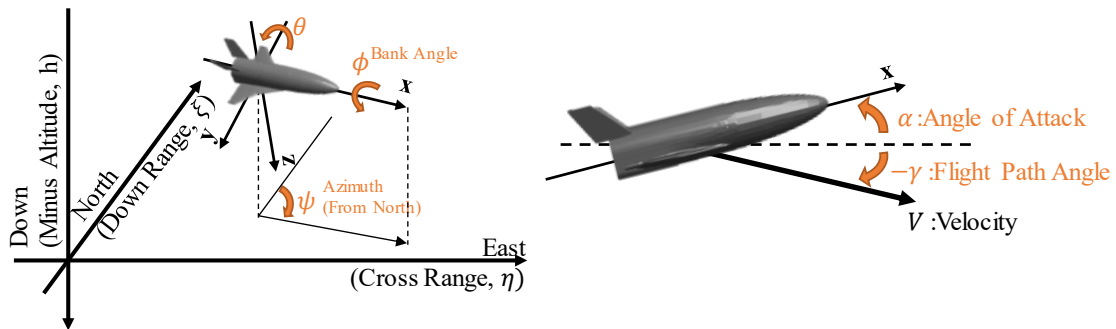


図 2.2-4 直行座標系の定義

ここで空力係数 C_L および C_D は、宇宙航空研究開発機構の宇宙科学研究本部が開発を進めた HIMES (Highly Maneuverable Experimental Space vehicle) のスケールモデルの風洞試験結果に基づく [16].

その空力係数 C_L および C_D について、図 2.2-5 に示す風洞試験結果から迎角 $0 \sim 20[\text{deg.}]$ ($0 \sim 0.3491[\text{rad}]$) の範囲においてマッハ数の条件毎に式(2.2-8)および式(2.2-9)のような迎角 α の関数として定義する.

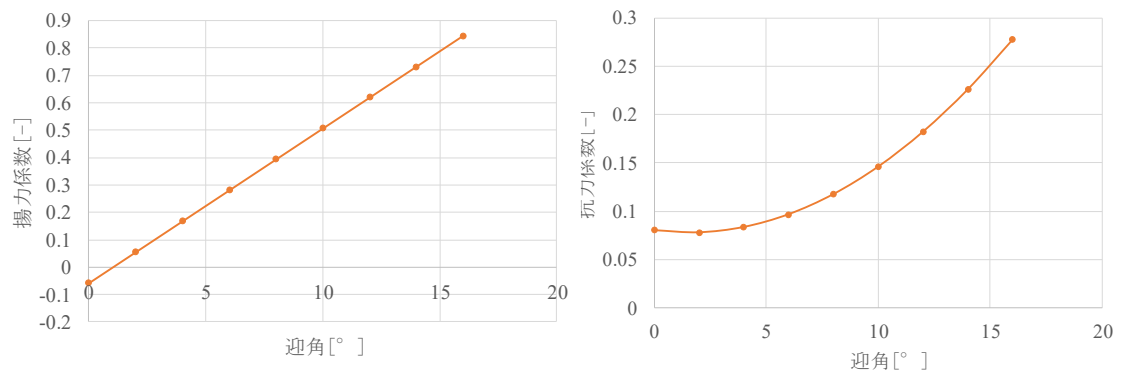


図 2.2-5 有翼ロケットの空力特性

$$C_L = 3.232585\alpha - 0.05748 \quad (2.2-8)$$

$$C_D = 3.146437\alpha^2 - 0.171525\alpha + 0.080523 \quad (2.2-9)$$

2.3. 動的分散遺伝的アルゴリズム

2.3.1. 遺伝的アルゴリズム

遺伝的アルゴリズム GA (Genetic Algorithm) とは、生命の進化を工学的にモデル化したアルゴリズムである[17].

GA は最適化変数を遺伝子として表現し、様々な遺伝子によって表される解(個体と呼ぶ)を複数用意し、この解集団の評価(適応度)を行い、交叉や突然変異といった遺伝子操作を行いながら解集団全体の適応度を向上させて1つの最適解(もしくは準最適解)に収束させていく手法である。GA は数値解析的な手法とは違い解が発散する心配が無く、大域的かつ柔軟な探索が行えるため、巡回セールスマン問題等非常に多くの問題に適応することが出来る。ただし、多数存在する個体全てに対し、適応度の計算を行う必要があるため、計算コストが嵩むというデメリットがある。また、乱数を用いている点および設定値の調整次第で探索性能が大きく変化する点から最適解に収束しない場合がある。

GA には遺伝子を2進数で扱うバイナリ GA と実数値をそのまま遺伝子として使用する実数値遺伝的アルゴリズム RCGA (Real-Coded Genetic Algorithm) がある。RCGA は最適化変数をそのまま使用することによりバイナリ GA と比較して効率よく親子対の形質を引き継ぐことができ、最適化に有利である。本論文では実数値 GA を使用するものとする。

図 2.3-5 に GA の処理の流れを示す。

GA の処理では、まず初期個体の生成を行い、適応度の計算、選択、交叉、突然変異の遺伝的処理から構成される世代個体を行う。世代交代を繰り返すことで個体集団の適応度を成長させていく。

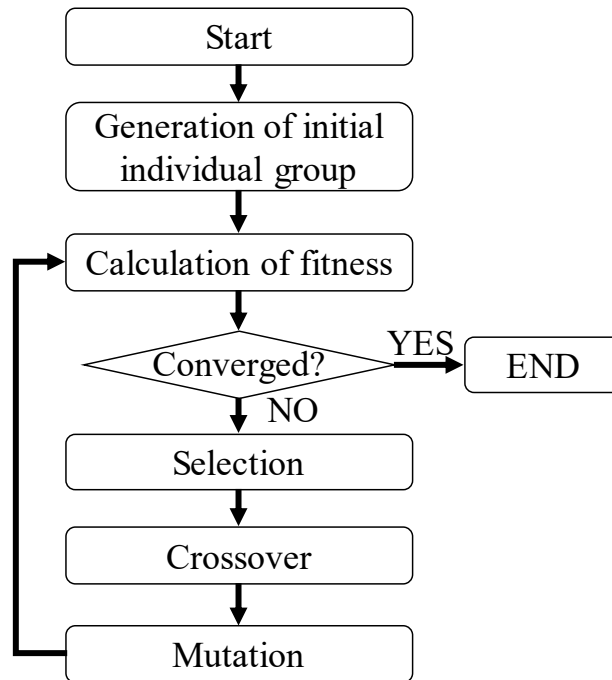


図 2.3-1 遺伝的アルゴリズムの処理の流れ

以下に各処理の詳細を述べる

＜初期個体集団生成 (Generation of initial individual group)＞

最適化における初期値の設定に当たる。GA では、初期個体を複数生成し、初期個体集団とする。各個体の各遺伝子をランダムもしくは一定の法則に従って決定する。初期個体集団の適応度が良好なほど、解の収束が早い。一方で、初期個体集団次第では局所解へ収束しやすくなってしまうこともある。

＜適応度計算 (Calculation of fitness)＞

各個体に対して個体の評価値である適応度の計算を行う。今期あの軌道生成問題では運動方程式を積分して得た軌道から適応度を計算する。GA の計算ではこの適応度を最大化もしくは最小化するように遺伝子処理を行う。この適応度の評価項目が複数存在し、それぞれ別に計算する手法の多目的最適化と、評価項目が単一、もしくは複数の評価項目を関数によって一つの評価項目にして計算する単目的最適化の 2 種類が存在し、本論文ではペナルティ法による単目的最適化を適用している。

＜選択 (Selection)＞

遺伝的処理における選択には、複製選択 (Selection for Reproduction) と生存選択 (Selection for Survival) の 2 種が存在する。複製選択は、後の処理である交叉

において子個体を生成する基準となる個体を選択することである．生存選択は交叉後の親個体および子個体から次世代の解集団とする個体を選ぶことである．適応度の扱い方を指定するため，解の収束性や安定性に大きく影響を及ぼす．選択の手法は世代交代モデルとして幾つか定義されており，Iterated Genetic Search (IGS), Steady State (SS), Elitist Recombination (ER), Minimal Generation Gap (MGG) 等が上げられる[18, 19]．本論文においては，従来研究より ER を用いている．

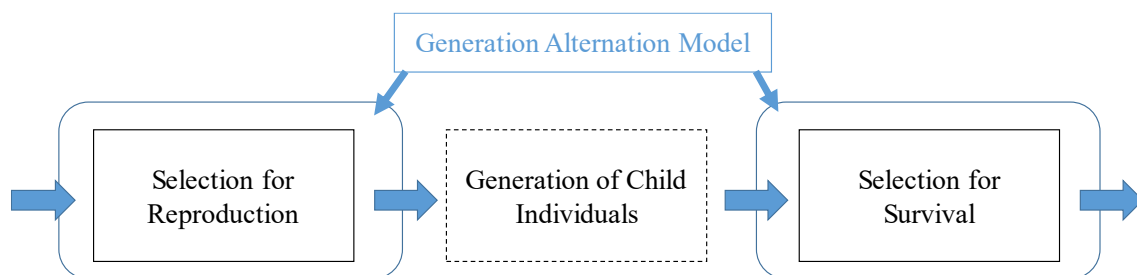


図 2.3-2 世代交代モデル

エリート選択 (ER) のイメージ図を図 2.3-3 に示す．複製選択はランダム非復元抽出により 2 個体ずつ選ばれる．生存選択では，最も適応度の高い 2 個体が次の世代として引き継がれる．エリート選択では，親個体集団を 2 個体ずつに分けてそれぞれ子個体を生成するため，生存選択が局所的になり，一つの局所解が急速に個体集団全体に広がる現象を防ぐ効果がある．

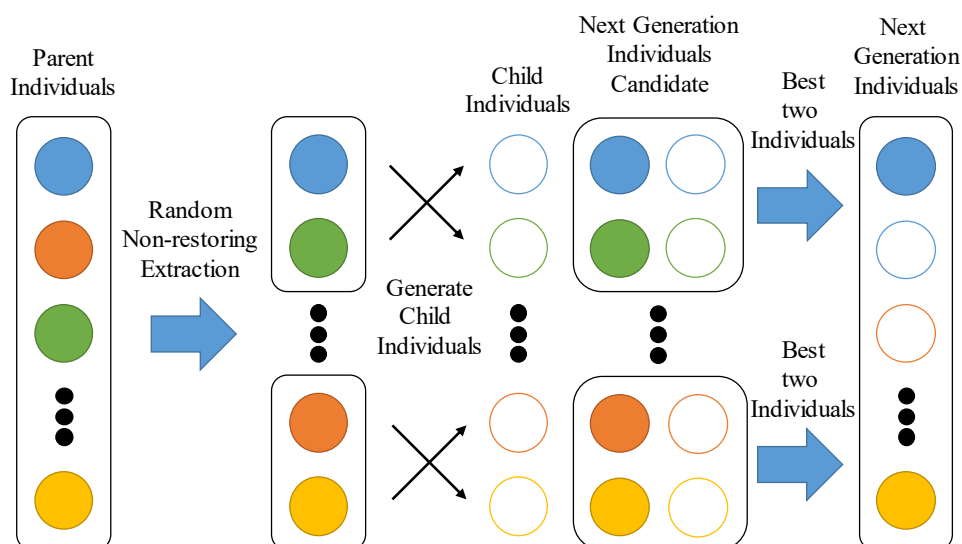


図 2.3-3 エリート選択 (ER)

<交叉 (Crossover)>

複製選択により選択された親個体より，子個体を生成する処理を交叉と呼ぶ．RCGA で用いられる交叉として，ブレンド交叉 (BLX)，シンプレックス交叉 (SPX) または正規分布交叉 (UNDX) 等が挙げられる[19]．本論文においては，ブレンド交叉を用いている．

2次元の遺伝子の場合のブレンド交叉のイメージを図 2.3-4 に示す．ブレンド交叉は正のパラメータ β_{BLX} を用いて親個体の角変数の区間 ΔP_i を両側に $\beta_{BLX}\Delta P_i$ だけ拡張した空間から，一様乱数に従ってランダムに子個体を生成する．ブレンド交叉では親個体が状態空間で離れて存在している場合には子個体も広い範囲から生成され，親個体が互いに近くに存在している場合には親個体の近傍に生成される．子個体の遺伝子 C_i は親個体の遺伝子 P_i^1, P_i^2 を用いて式(2.3.1)で求められる．

$$C_i = \text{rand}[\min(P_i^1, P_i^2) - \alpha_{BLX}\Delta P_i, \max(P_i^1, P_i^2) + \alpha_{BLX}\Delta P_i] \quad (2.3-1)$$

ここで， $\text{rand}[\lambda_1, \lambda_2]$ は区間 $[\lambda_1, \lambda_2]$ の一様乱数を示す．

また， ΔP_i は親個体の遺伝子より

$$\Delta P_i = |P_i^1 - P_i^2| \quad (2.3-2)$$

で表される．

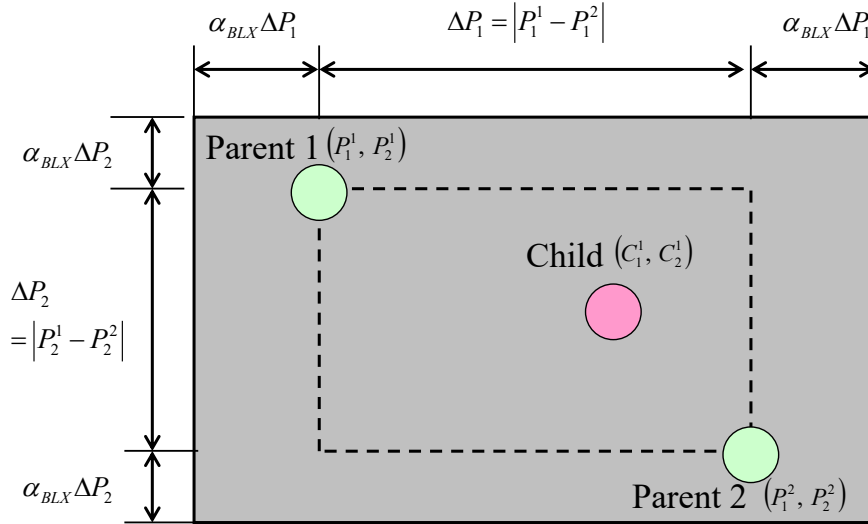


図 2.3-4 ブレンド交叉

<突然変異 (Mutation) >

突然変異では、ある一定の確率で遺伝子を変化させることによって、親個体を用いて行う交叉では生成できない全く異なった個体を生成する。このことより、個体集団の多様性の維持や局所解からの脱出を行うことができる。

RCGA では、一様突然変異や境界突然変異等が突然変異の手法として挙げられる。本論文では一様突然変異を用いている。

一様突然変異の例を図 2.3-5 に示す。遺伝子毎に一様乱数を用いて、実行可能領域に実数値を生成する。

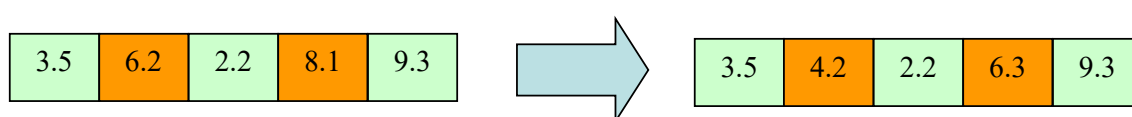


図 2.3-5 一様突然変異

2.3.2.分散遺伝的アルゴリズム

遺伝的アルゴリズムで解を成長させるためには解の多様性が重要である。一般的に母集団を一つのグループとして解成長させた場合、局所解に陥ってしまい個体はその局所解に収束して成長しなくなることがある。

これを防ぐのは突然変異であるが、そのほかの対策としては母集団を分割してそれぞれで解成長させることが考えられる。これは他の島から移住処理により全く異なった特徴をもつ個体を島内に追加することで多様性を維持し局所解に陥ることを防ぐものである。分散遺伝的アルゴリズム DGA (Distributed Genetic Algorithm) とは、GA における母集団をいくつかの「島」と呼ばれるサブ母集団に分割し、それぞれの島で GA の計算を行うものである[10]。また数世代毎に各島の一部の個体を入れ替える「移住」処理を行うことで、島内の解探索が停滞する状態を防ぐ。移住する個体の数は移住率と呼ばれる割合により設定する。図 2.3-6 に DGA の概略を示す。

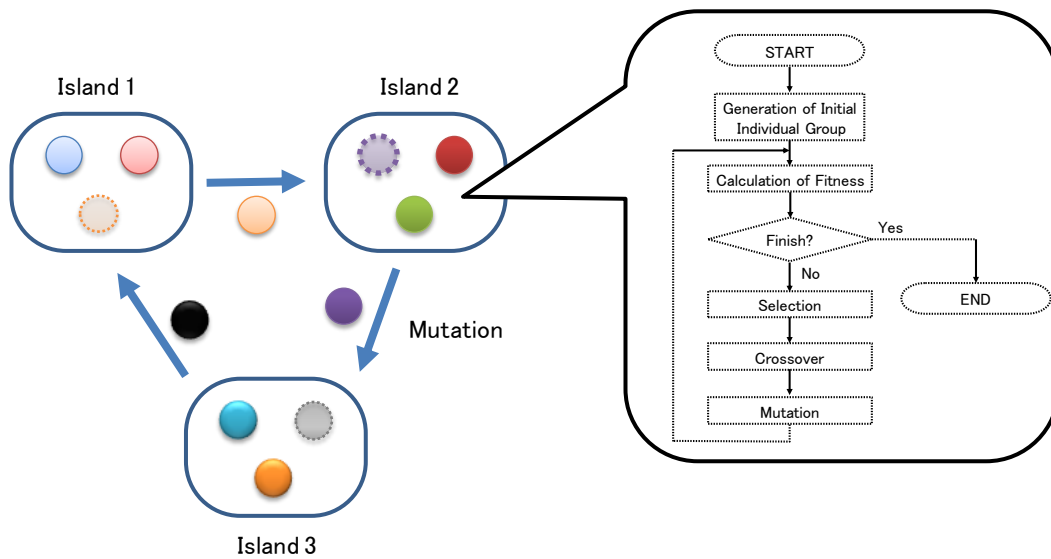


図 2.3-6 DGA アルゴリズム

移住についても幾つかのモデルが考えられている．移住モデルとしては図 2.3-7 に示すような隣接する島間での移住を行う踏み石型モデル(Stepping Stone Population Model)やランダムな母集団間で移住を行うランダム移住型モデル(Randomized Migration Island Model)がある[12,13].

踏み石型モデルは，個体の移住が隣接する島に一方向に移住が行われていく．このため，ある島内に良好な個体があったとしてもその個体が移住により全体に広がっていく可能性は非常に低く最適解への収束は遅い．それに対してランダム移住型モデルは全体の島をランダムに順序付けて個体の移住を行う．このため，良好な個体は移住によりさまざまな島に広がり易く最適解への収束は早いとその分局所解への収束に陥る可能性も高くなる．

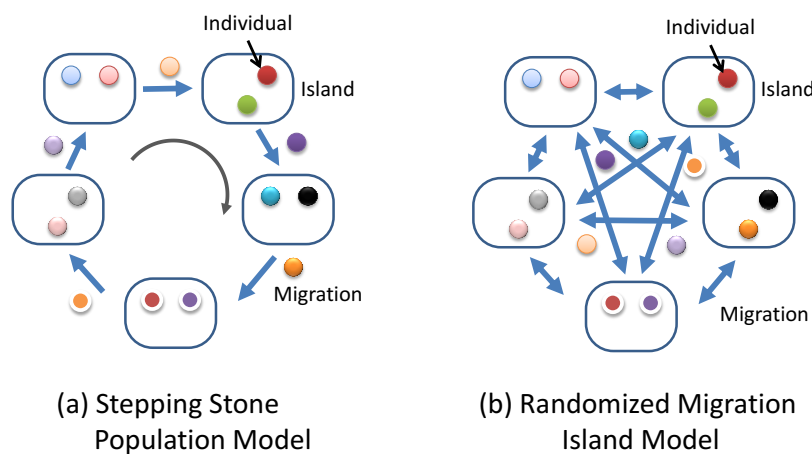


図 2.3-7 DGA の移住モデル

2.3.3.動的分散遺伝的アルゴリズム

DGA を用いることにより，島毎に異なる種類の個体が進化する，すなわち解の多様性を維持して複数の解を同時に生成できることが解った．しかし，島の間で個体の移住を可能とする操作を行っても，逆に個体の進化が妨げられてしまう場合があること，島に属する個体数や島の数そのものの設定方法には何も指針が得られないという遺伝的アルゴリズム適用の根幹に関わる課題が浮き彫りとなった．この課題を克服するために，進化の状況に応じて母集団を適宜分割または統合しながら，島に属する個体の進化を促すという新しい概念として動的分散遺伝的アルゴリズム DynDGA (Dynamically Distributed Genetic Algorithm) が提案された．DynDGA は，従来の DGA のように島の数やその島に属する個体数を固定せず，各個体の特性を端的に表すような特徴量を定義し，その特徴量によって個体間の非類似性を識別，非類似性から動的に解集団を島へ分割および統合しながら DGA の計算を行っていく．

図 2.3-8 に DynDGA の処理のイメージを示す．まず，通常の GA と同様に初期個体を生成し，ある程度世代を進めて解の成長を促す．ある程度解集団が成長すると，それぞれの個体に特徴が現れる．その特徴毎に個体を島へと分割し，島毎に遺伝子処理を行う．またしばらく世代を進めると，各島の解が変化するため，また特徴毎に島へと分割または統合を行う．局所解へ陥らないように，通常の GA と同様の突然変異の他に，島間の移住処理を行い，全体において会の多様性を維持しながら大域的な探索が行えるようになっていく．これらを繰り返すことによって，最終的に DynDGA は複数の最適解（または準最適解）を導出することが出来る．

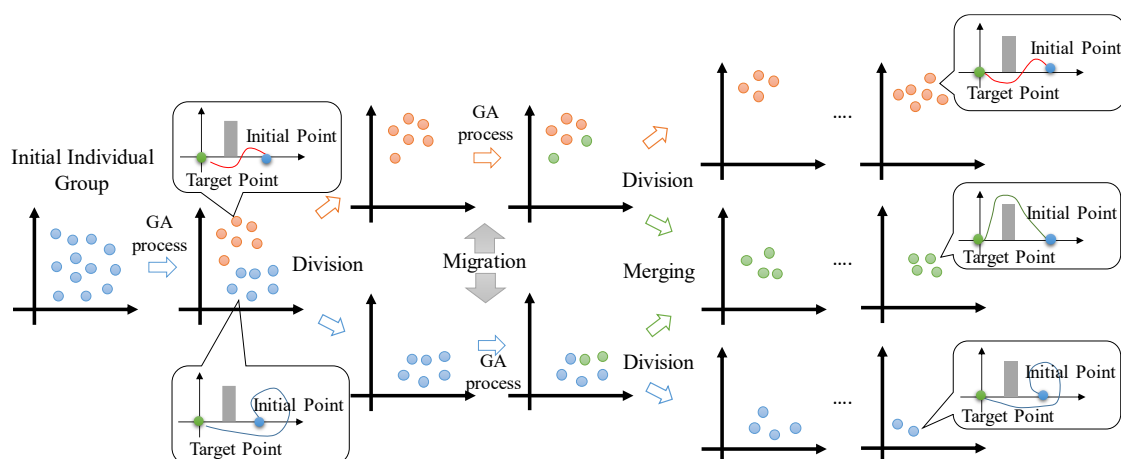


図 2.3-8 DynDGA の処理イメージ

2.3.4. 遺伝子の設定

本論文では、迎角 α 、バンク角 ϕ の2つの制御入力を最適化する。しかし、従来よく用いられるように制御入力を時間についての離散データとすると最適化変数の数が膨大になり、計算コストが嵩む。そこで、式(2.3-3)および式(2.3-4)のように制御入力をフーリエ級数で表現し、その係数(式内赤文字部)を遺伝子とすることで、設計変数を大幅に削減する手法が提案された。フーリエ級数を3次とした場合、各個体の遺伝子の数は合計18個の実数となる。なお、 t は時間を表す。

$$\alpha(t) = \frac{a_{0\alpha}}{2} + \sum_{n=1}^3 \{a_{n\alpha} \cos(n\omega_{n\alpha}t) + b_{n\alpha} \sin(n\omega_{n\alpha}t)\} \quad (2.3-3)$$

$$\phi(t) = \frac{a_{0\phi}}{2} + \sum_{n=1}^3 \{a_{n\phi} \cos(n\omega_{n\phi}t) + b_{n\phi} \sin(n\omega_{n\phi}t)\} \quad (2.3-4)$$

ただし、各制御入力には次のような角度制限を設ける

$$\begin{cases} \alpha(t) \leq \frac{\pi}{36} [\text{rad}] (= 5[\text{deg.}]) & \alpha(t) = \frac{\pi}{36} [\text{rad}] \\ \alpha(t) \geq \frac{\pi}{12} [\text{rad}] (= 15[\text{deg.}]) & \alpha(t) = \frac{\pi}{12} [\text{rad}] \end{cases} \quad (2.3-5)$$

$$\begin{cases} \phi(t) \leq -\frac{\pi}{3} [\text{rad}] (= -60[\text{deg.}]) & \phi(t) = -\frac{\pi}{3} [\text{rad}] \\ \phi(t) \geq \frac{\pi}{3} [\text{rad}] (= 60[\text{deg.}]) & \phi(t) = \frac{\pi}{3} [\text{rad}] \end{cases} \quad (2.3-6)$$

しかし、上記式のようなフーリエ級数によって最適化を行った場合、各遺伝子の絶対値が大きくなりすぎると、角度制限によって同じ値ばかりを出力することとなり、遺伝子処理によって遺伝子が多少の変化をした場合でも、最終的に出力される遺伝子に変化が起こらない可能性がある (図 2.3-9)。

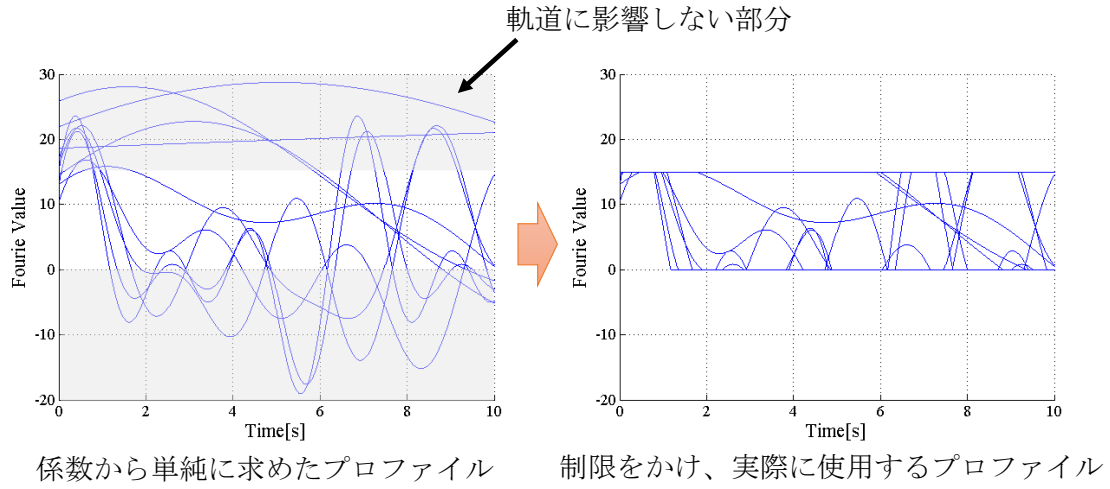


図 2.3-9 フーリエ級数の数値的制限

そこで、各遺伝子の値の範囲を $-1 \leq a_n, b_n, \omega \leq 1$ とし、式(2.3-7)～式(2.3-10)のような式を用いて各制御入力の値を表現するように改良した。

$$\theta_{\alpha}(t) = \frac{a_{0\alpha}}{2} + \frac{1}{3} \left(1 - \left| \frac{a_{0\alpha}}{2} \right| \right) \sum_{n=1}^3 \left\{ \frac{a_{n\alpha} \cos(n\omega_{\alpha}t) + b_{n\alpha} \sin(n\omega_{\alpha}t)}{\sqrt{a_{n\alpha}^2 + b_{n\alpha}^2}} \right\} \quad (2.3-7)$$

$$\theta_{\phi}(t) = \frac{a_{0\phi}}{2} + \frac{1}{3} \left(1 - \left| \frac{a_{0\phi}}{2} \right| \right) \sum_{n=1}^3 \left\{ \frac{a_{n\phi} \cos(n\omega_{\phi}t) + b_{n\phi} \sin(n\omega_{\phi}t)}{\sqrt{a_{n\phi}^2 + b_{n\phi}^2}} \right\} \quad (2.3-8)$$

$$\alpha(t) = \theta_{\alpha}(\alpha_{\max} - \alpha_{\min}) + \alpha_{\min} \quad (2.3-9)$$

$$\phi(t) = \theta_{\phi}(\phi_{\max} - \phi_{\min}) + \phi_{\min} \quad (2.3-10)$$

ただし、 α_{\max} は迎角の角度制限の最大値側(= $\frac{\pi}{12}$ [rad]) であり、 α_{\min} は迎角角度制限の最小値側(= $\frac{\pi}{36}$ [rad])、同様に ϕ_{\max} はバンク角角度制限の最大値側(= $\frac{\pi}{3}$ [rad]) であり、 ϕ_{\min} はバンク角角度制限の最小値側(= $-\frac{\pi}{3}$ [rad]) である。

この式を用いて表されるフーリエ級数の例を図 2.3-10 に示す。

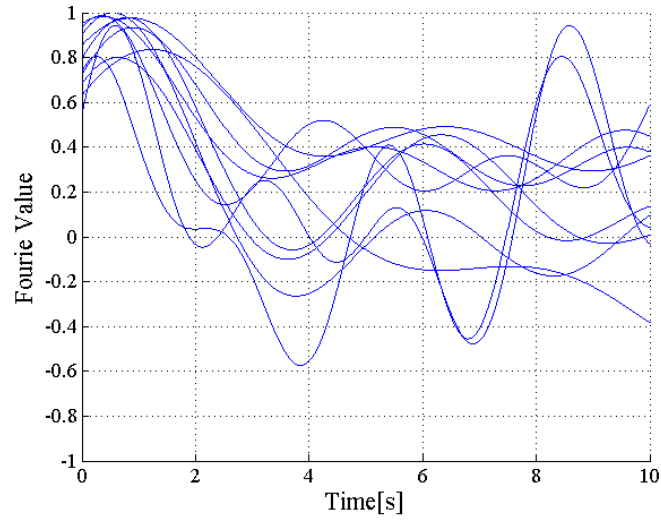


図 2.3-10 正規化したフーリエ級数の例

これによって、係数の変化が必ず制御入力の変化に影響するようになり、解の探索効率と収束性が向上した。

一方で、図 2.3-11 に示す様な境界付近の解を最適解とする最適化問題の場合、探索性能が非常に低下してしまうことも解った。

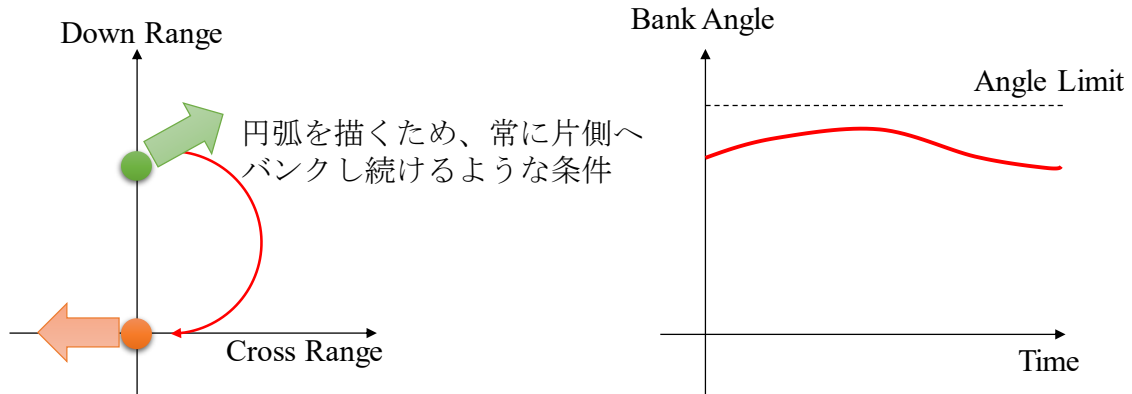


図 2.3-11 最適解が境界条件付近の場合の例

そこで、軌道の最適化に特化した制御入力の表現方法として、フーリエ級数を式(2.3-11)～式(2.3-13)のように拡張し、解の探索性能向上を図った。

$$\theta(t) = \text{sign}(a_0) a_0^2 + \frac{1}{3} (1 - |a_0^2|) \sum_{n=1}^3 \left\{ a_n \cos \left(n \left(\frac{\omega}{2} + 0.5 \right)^6 t + b_n \pi \right) \right\} \quad (2.3-11)$$

$$\alpha(t) = \theta_\alpha (\alpha_{\max} - \alpha_{\min}) + \alpha_{\min} \quad (2.3-12)$$

$$\phi(t) = \theta_\phi (\phi_{\max} - \phi_{\min}) + \phi_{\min} \quad (2.3-13)$$

これにより、境界付近の解の探索も行いやすくなり、全体の解探索性能が向上した（図 2.3-12）.

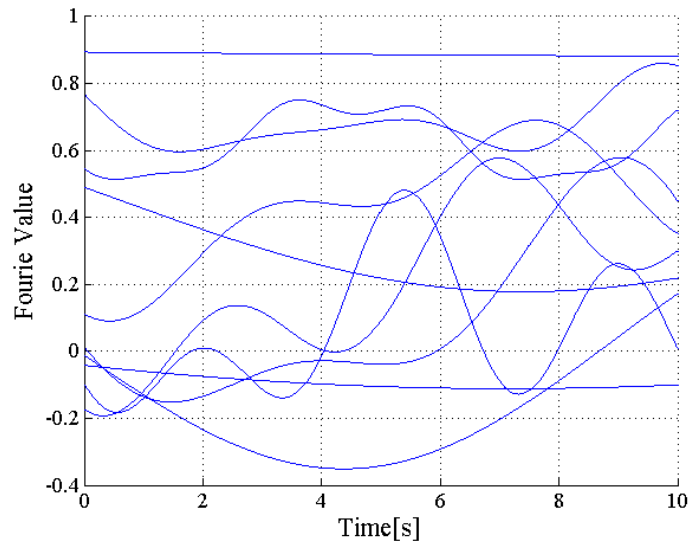


図 2.3-12 軌道最適化に特化したフーリエ級数の制御入力表現

2.3.5.適応度

一般的な GA を用いた最適化においては、解が必ず満たしていなければならない項目を表す制約条件と出来るだけ評価を高めたい項目を表す目的関数が用いられる．本論文における軌道の最適化では、図 2.3-13 に示すような項目が制約条件および目的関数となる．

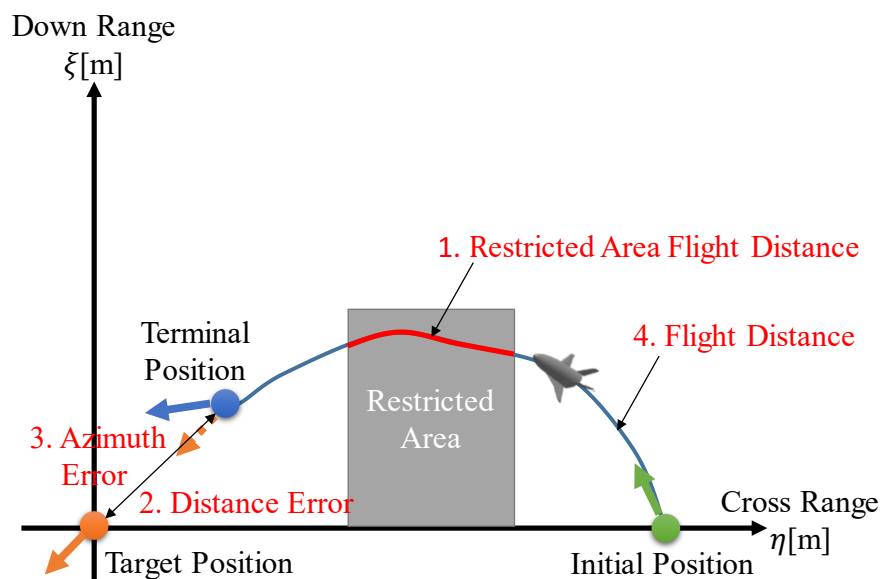


図 2.3-13 適応度評価項目イメージ

ここで最適化したいパラメータが複数存在するため、多目的最適化として扱うことも可能であるが、飛行禁止区域飛行距離および到達地点距離誤差、到達地点方位角誤差のような制約条件と飛行時間といった目的関数を重み付けの和で表すペナルティ法とよばれる手法を用いて、式(2.3-14)に示す様な関数を適応度として用いて計算を進める。本誘導システムにおける最適化ではこの適応度を最大化するように遺伝的処理を行う。

$$f = \frac{s}{w_t t_f + w_\xi |\xi_T - \xi_f| + w_\eta |\eta_T - \eta_f| + w_h |h_T - h_f| + w_\psi |\psi_T - \psi_f|^2 + w_R R} \quad (2.3-14)$$

ここで w は各重み係数、 R は飛行禁止区域飛行距離、添字 T は目標位置における値、添字 f は軌道終端位置における値を示す。 s は $s = 1000$ とし、各重み係数はシミュレーションを何度か行う中で、各項の影響度が一定となるように決定する。これらの値は運動方程式を積分計算して求めていく。これにより、飛行禁止区域を回避し、目標地点と軌道終端位置のそれぞれの項の誤差が小さいものほど適応度の値が大きくなる。

2.3.6. 特徴量

DynDGA を利用するには、各軌道（個体）の特徴量を設定する必要がある。この特徴量を元に個体間の距離（非類似度）を計算することによって個体の分類を行うため、使用する特徴量によって、分類性能が大きく変化する。先行研究より、本論文では運動方程式を積分して求めた軌道データを元に特徴量を算出する方法を用いている[9]。図 2.3-14 に軌道データを元に特徴量を抽出するイメージを示す。軌道のダウンレンジとクロスレンジの履歴を飛行時間の関数として扱い、等分割して順番に並べることによって、等分割された時間におけるダウンレンジとクロスレンジの値を特徴量として定義した。

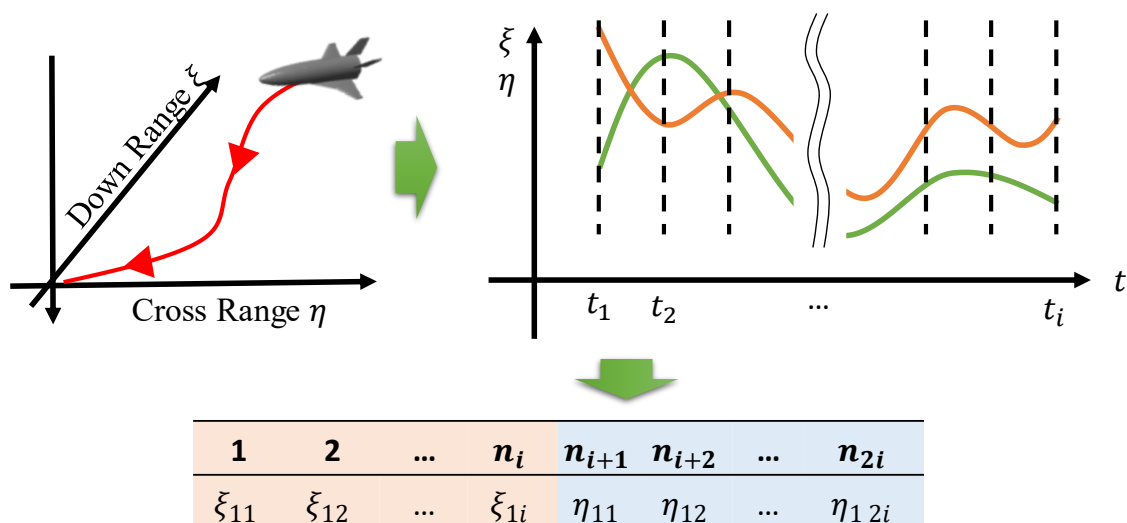


図 2.3-14 ダウンレンジとクロスレンジを時間の関数とする特徴量の定義

2.3.7. 個体間の距離の算出と非類似度

母集団を類似の個体によってサブ母集団に分割あるいは統合するには、図 2.3-14 に定義する特徴量を元に、個体間の距離（非類似度）を定義する必要がある。この個体間の距離が小さいと類似性が高く、逆に大きいと類似性が低いこととなる。その類似性を測るための距離の定義式には幾つか種類が存在するが、従来研究より式(2.3-15)で定義されるユークリッド平方距離を本論文では用いている。

ユークリッド平方距離は、ユークリッド空間内で最も基本的な距離定義であるユークリッド距離（図 2.3-15）は 3 次元空間におけるユークリッド平方距離の例を表す）の平方根計算を省略した形となっている。このため、平方根処理を苦手とするマイクロコンピュータなどの計算負荷軽減を目的に、実装時に使用されることが多い。

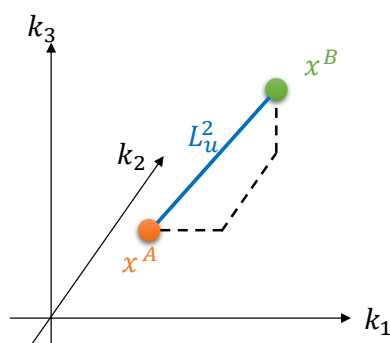


図 2.3-15 ユークリッド（平方）距離の例

$$L_u = \sum_{k=1}^{n_f} (x_{ik}^A - x_{jk}^B)^2 \quad (2.3-15)$$

ここで x_{ik}^A はクラスタ A の個体 i の k 次元目の特徴量, x_{jk}^B はクラスタ B の個体 j の k 次元目の特徴量を意味する. また, n_f は特徴量の次元数であり, L_u がユークリッド平方距離である. なお, クラスタとはサブ母集団のことである.

図 2.3-14 より式(2.3-15)を用いた非類似度の計算例は式のようになる.

$$d_{12} = (\xi_{11} - \xi_{21})^2 + \dots + (\xi_{1i} - \xi_{2i})^2 + (\eta_{11} - \eta_{21})^2 + \dots + (\eta_{1i} - \eta_{2i})^2 \quad (2.3-16)$$

ここで d_{12} は個体 1 と個体 2 の非類似度で ξ_{ik} は個体 i の k 番目のダウンレンジを表す. 同様に η_{ik} は個体 i の k 番目のクロスレンジを表す.

2.3.8.階層的クラスタリング

階層型クラスタリングとは, 非類似度の小さい (すなわち類似性の高い) 個体の集合体としてのクラスタ同士を順番に結合し, さらに大きなクラスタを形成していく手法である. ここで, 非類似度はクラスタ間の距離を示す.

クラスタリングの手順は以下の通りである.

- ① 個体数 N 個分のクラスタを用意し, 各クラスタには, 1 個ずつ個体が割り振られる.
- ② クラスタ間の非類似度について, 全ての組み合わせの非類似度行列 \mathbf{d} を作成する.

$$\mathbf{d} = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1N} \\ d_{21} & d_{22} & \dots & d_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1} & d_{N2} & \dots & d_{NN} \end{bmatrix}$$

ただし, 対角成分 $d_{11}, d_{22}, \dots, d_{NN}$ は 0 である. また, $d_{ij} = d_{ji}$ の対称行列である.

- ③ 非類似度行列から最小の非類似度となるクラスタの組み合わせ d_{ij} ($i \neq j$)を探し, その 2 つのクラスタを結合する. これにより, クラスタ数は N-1 となる.
- ④ クラスタ数が 1 になるまで②③を繰り返す.

階層的クラスタリングでは, 横軸を個体, 縦軸を結合時の非類似度とし, 樹形図 (デンドログラム) を構成する. 最小の非類似度である結合対象のクラスタ同

士が隣になるように個体を移動させ、グラフを見やすくさせる．本論文における母集団の分割に当たっては，デンドログラムを用いて図 2.3-16 に示す様に，ある固定した非類似度となるところでクラスタに分割する方法を採用している．

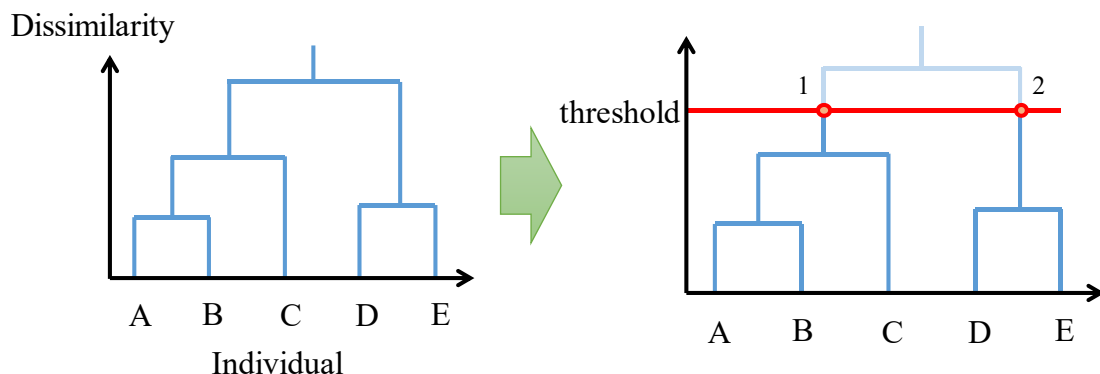


図 2.3-16 階層的クラスタリングにおけるデンドログラム

階層的クラスタリングにおいて，各クラスタにおける非類似度の計算には，いくつかの手法があり，「組み合わせ手法」として定義されている．本論文では，Ward 法 (Ward's Method) を用いて非類似度の計算を行っている．

Ward 法のイメージを図 2.3-17 に示す．

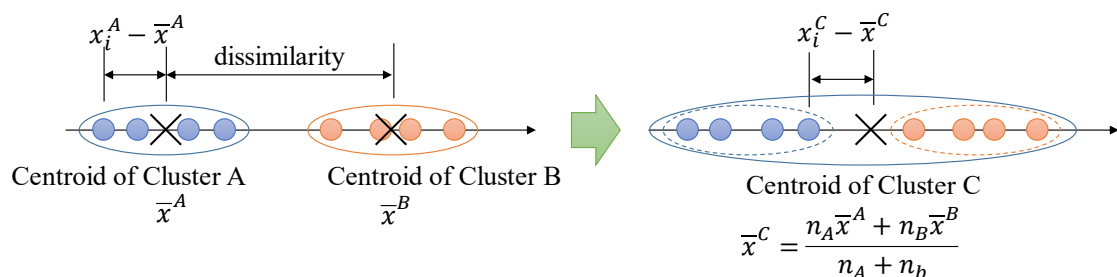


図 2.3-17 Ward's Method

Ward 法は 2 つのクラスタ A と B を結合したときのクラスタ C 内のデータの平方和の変化を非類似度として定義する．

$$d_{AB} = \sum_{i=1}^P \left\{ \sum_{j=1}^{n_A+n_B} (x_{ij}^C - \bar{x}_i^C)^2 - \sum_{j=1}^{n_A} (x_{ij}^A - \bar{x}_i^A)^2 - \sum_{j=1}^{n_B} (x_{ij}^B - \bar{x}_i^B)^2 \right\} \quad (2.3-17)$$

ここで

$$\bar{x}_i^A = \frac{1}{n_A} \sum_{j=1}^{n_A} x_{ji}^A \quad (2.3-18)$$

$$\bar{x}_i^B = \frac{1}{n_B} \sum_{j=1}^{n_A} x_{ji}^B \quad (2.3-19)$$

$$\bar{x}_j^C = \frac{1}{n_C} \sum_{i=1}^{n_C} x_{ji}^C = \frac{n_A \bar{x}_i^A + n_B \bar{x}_i^B}{n_A + n_B} \quad (2.3-20)$$

この平方和の変化は対象の 2 つのクラスタの重心 \bar{x}_i^A , \bar{x}_i^B とそれぞれのクラスタ内の個体数 n_A , n_B を用いて以下の式で表すことが出来る．

$$d_{AB} = \frac{n_A n_B}{n_A + n_B} \sum_{i=0}^P (\bar{x}_i^A - \bar{x}_i^B)^2 \quad (2.3-21)$$

なお各非類似度の更新に際して、全ての組み合わせを上記の定義式に従って再計算する必要は無く、結合前の非類似度を元に計算することが出来る[20].

いま、クラスタ i (個体数 n_i) とクラスタ j (個体数 n_j) を結合して新しいクラスタ k (個体数 $n_k = n_i + n_j$) を生成したとする．このとき、結合したクラスタ k と他のクラスタ h との非類似度 d_{kh} の計算には、結合前の各クラスタ間の距離 d_{hi} , d_{hj} および d_{ij} を用いて以下の式から求めることが出来る．なお、個体間の距離にはユークリッド平方距離を用いている場合である．

$$d_{kh} = \frac{n_h + n_j}{n_h + n_k} d_{hi} + \frac{n_h + n_j}{n_h + n_k} d_{hj} + \frac{n_h}{n_h + n_k} d_{ij} \quad (2.3-22)$$

2.3.9. DynDGA を用いた誘導軌道生成

DynDGA を用いた誘導軌道生成アルゴリズム実証のため、コンピュータを用いて軌道生成シミュレーションを行った．

計算条件を表 2.3-1 に示す．また、飛行領域の概略を図 2.3-18 に示す．

表 2.3-1 DynDGA 軌道生成シミュレーション条件

Number of Individuals	100
Number of Evolutions	800
Crossover Method	Blend Crossover (BLX-0.5)
Genetic Operator	Elitist Recombination
Mutation Rate	0.2
Integration Method for Equation of Motion	4 th Runge-Kutta Method
Integration Time Step	0.5
Migration Rate	0.1
Migration Interval	20
Dissimilarity Threshold	8×10^6

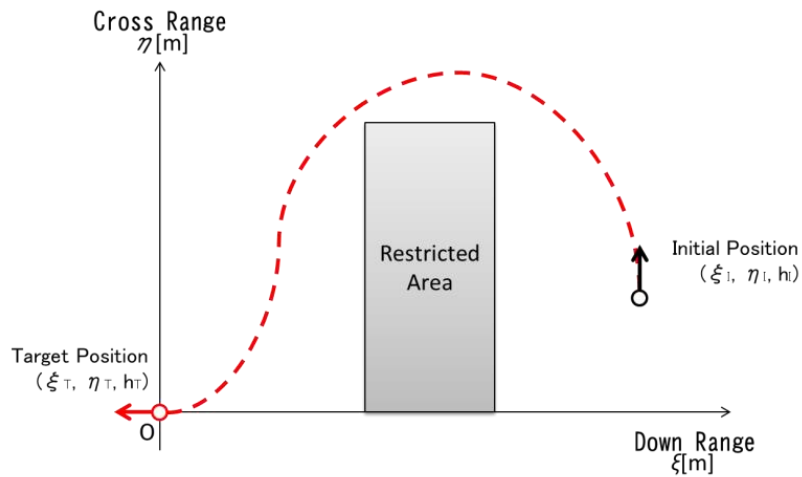


図 2.3-18 シミュレーションの飛行領域

シミュレーション結果を図 2.3-19 から図 2.3-25 に示す.

飛行禁止区域を避けながら目標地点へ到達できる軌道が複数生成できているのが確認できる. 目標地点付近に到達する軌道において, 軌道終端位置における方位角は誤差が残ってしまっている.

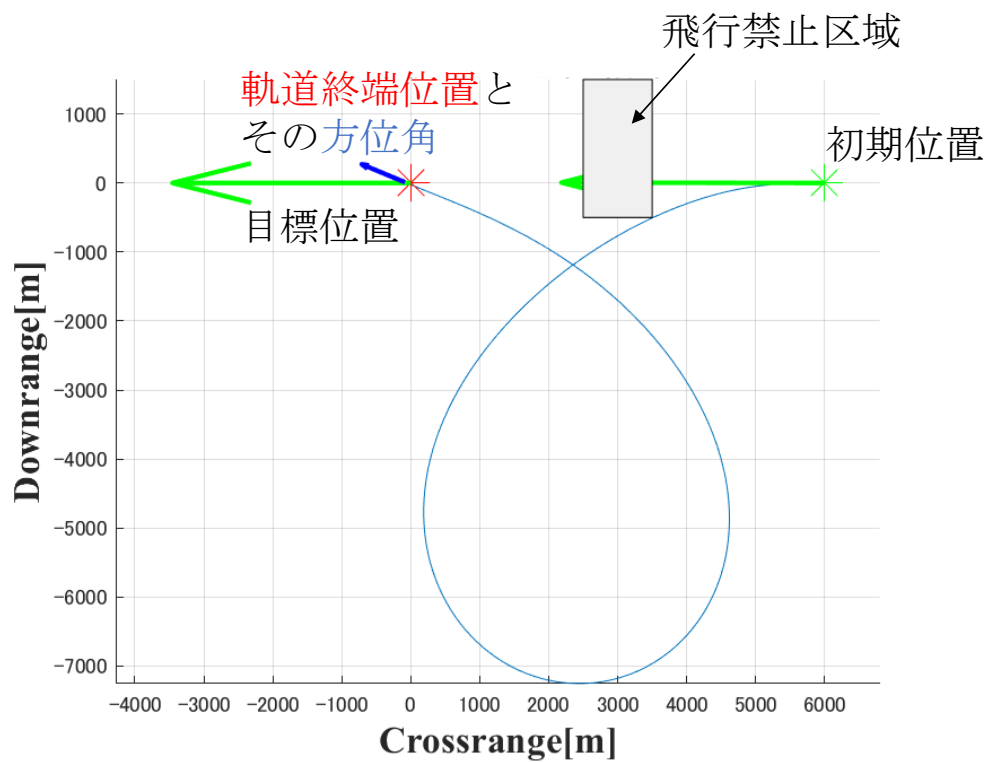


図 2.3-19 DynDGA を用いた誘導軌道最適化結果(最適軌道)

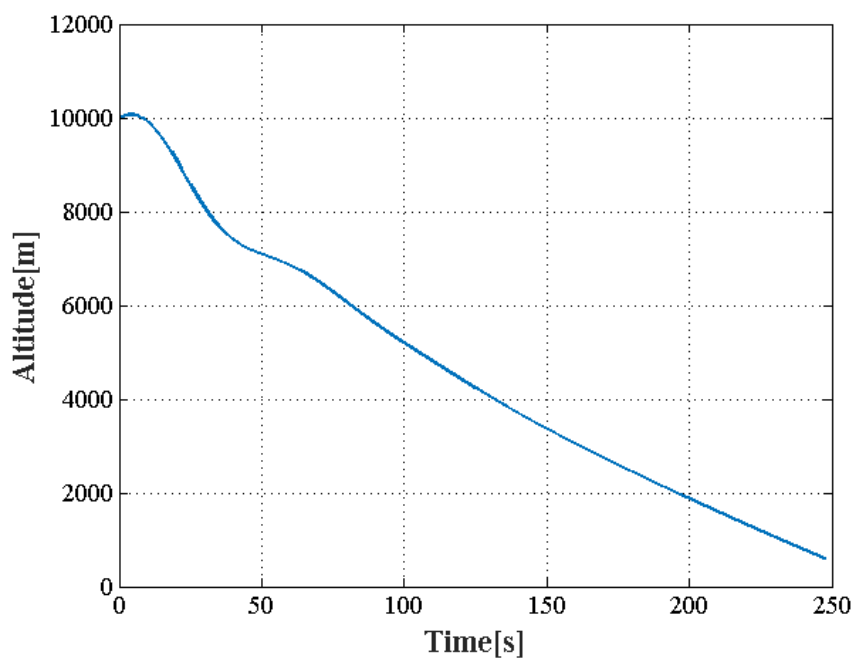


図 2.3-20 DynDGA を用いた誘導軌道最適化結果(高度プロファイル)

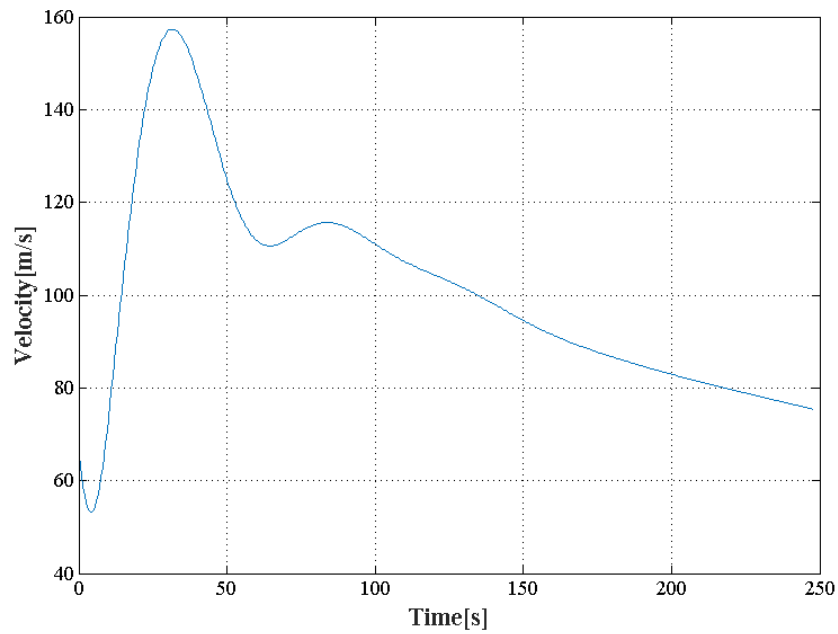


図 2.3-21 DynDGA を用いた誘導軌道最適化結果(速度プロファイル)

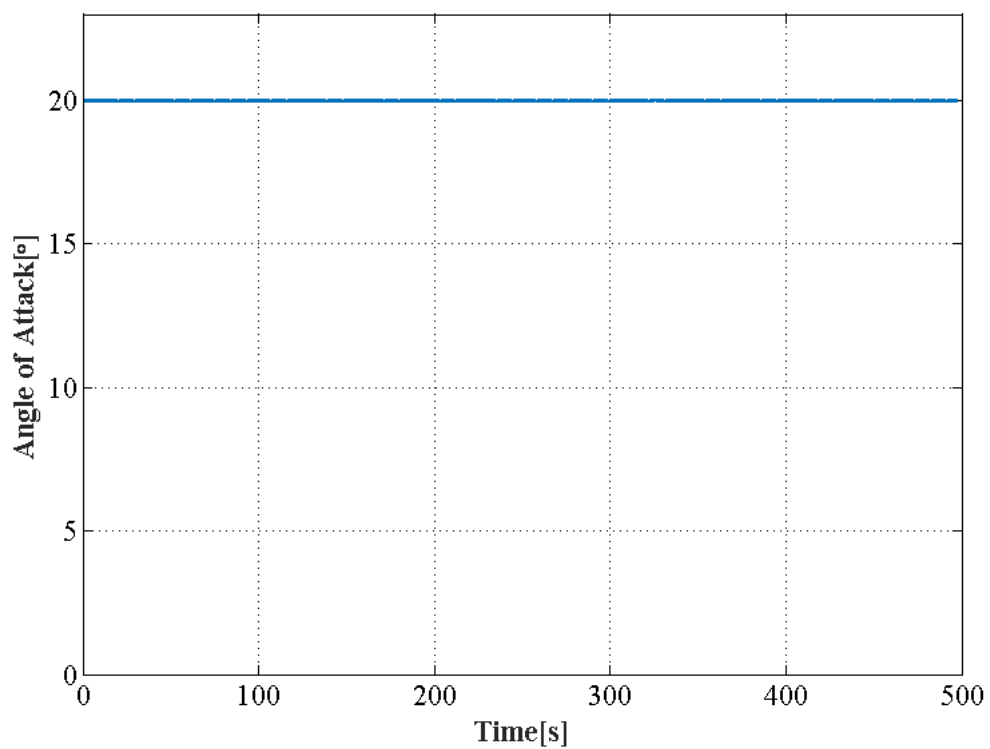


図 2.3-22 DynDGA を用いた誘導軌道最適化結果(迎角プロファイル)

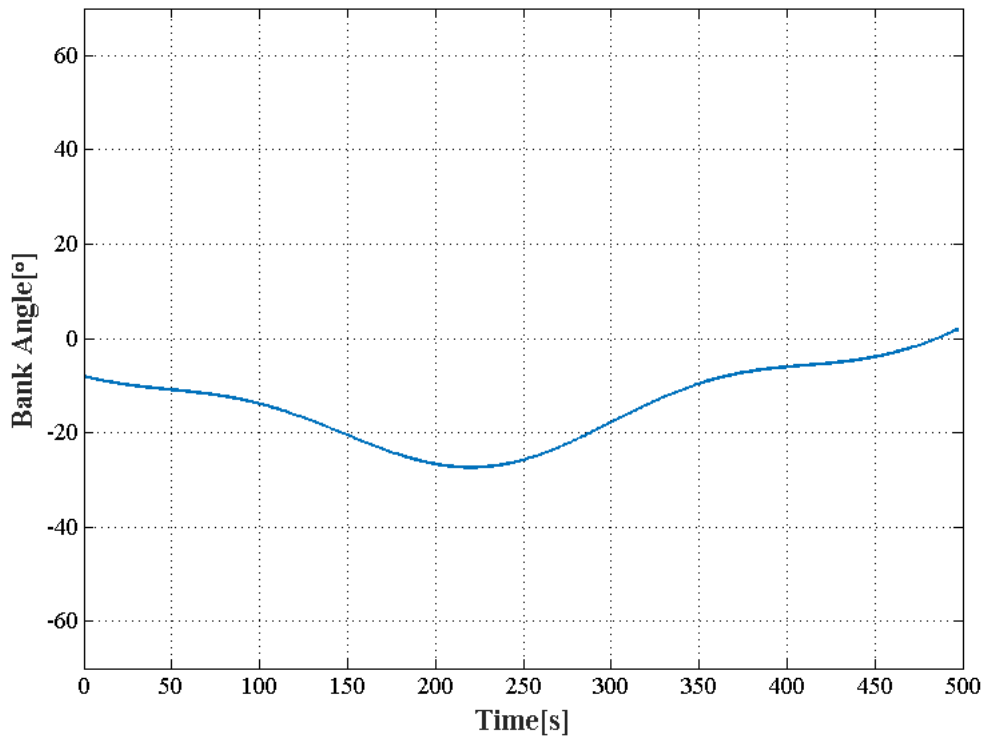


図 2.3-23 DynDGA を用いた誘導軌道最適化結果(バンク角プロファイル)

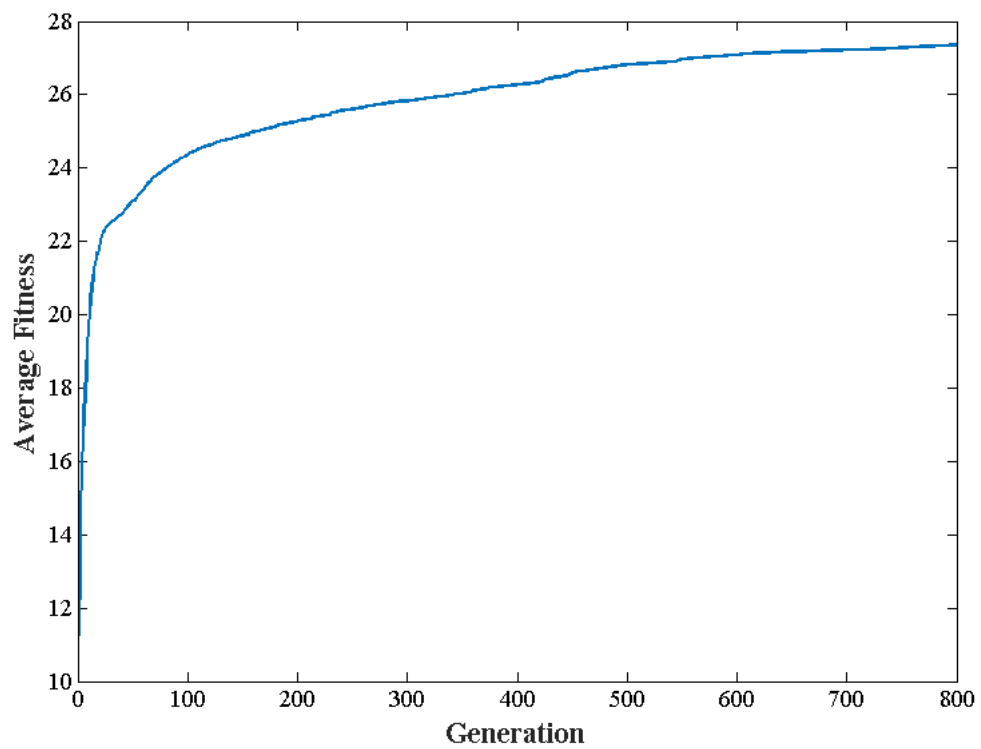


図 2.3-24 DynDGA を用いた誘導軌道最適化結果(平均適応度)

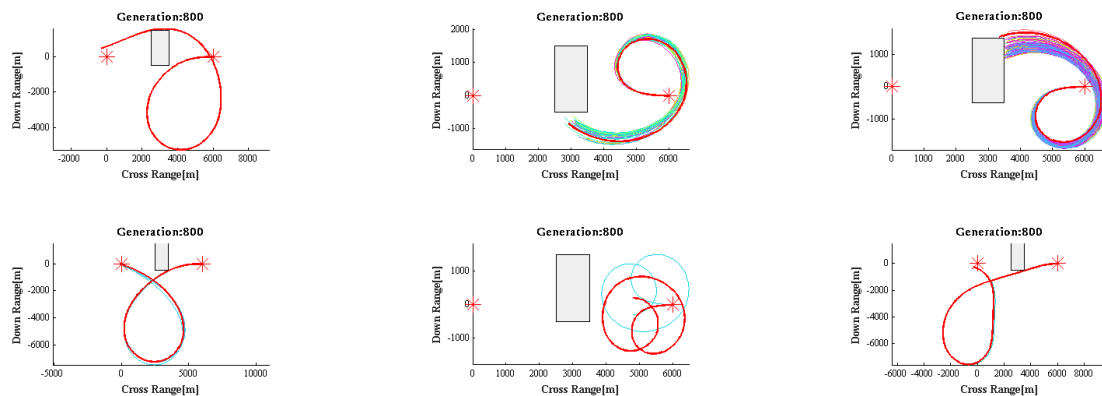


図 2.3-25 DynDGA を用いた誘導軌道最適化結果(最終世代の軌道)

2.3.10. おわりに

本章では、DynDGA による最適軌道の探索を行い、遺伝的アルゴリズムをもちいた軌道生成が柔軟な解探索手法であることを述べた。また、同時に複数種類の最適軌道を算出可能であり、ユーザーが飛行する軌道を選ぶことが出来ると同時に、不測の事態において一つの最適解が使用不能になっても、別の誘導軌道から安全に目標地点へ滑空帰還飛行を行えることを示せた。

一方で、計算コストが通常の GA と比較して、個体数が多く必要であること、成長するまでに必要とする世代数が多く必要であること、クラスタリングなどの処理を行う必要があることなどから、より嵩んでしまっていることが解った。

また、予め設定しておかなければならない設計値の数も、クラスタリングの閾値や移住処理の確率、間隔などと増加し、計算コストが嵩んでしまっているために試行錯誤的に調整する必要がある適応度の重み係数やクラスタリングの閾値などの決定にもかなりの時間が必要となってしまうことがわかった。

本誘導システムを実用化するに当たり、試行錯誤的に決定する必要がある設定値と計算コストといった 2 つの課題について、新たな提案を次章で行う。

第3章 ファジィ集合による階層的適応度を応用した 動的分散遺伝的アルゴリズムを用いた軌道生 成と係数自動調整手法の提案

3.1. はじめに

第2章において DynDGA を用いた最適軌道生成システムの処理とその誘導シミュレーション結果を示した。そして、DynDGA によって通常の最適化とは異なり、複数の最適解（準最適解）を求めることが可能であることがわかった。一方で通常の GA と比較して、計算コストが嵩むこと、試行錯誤的に調整が必要な設定値が多いこともわかった。

本章では、試行錯誤的に調整を行っていた設定値の一つである適応度の重み係数について、最適化前に指定した優先度に従って、適応度の成長が行われるように最適化を進めるファジィ集合による階層的適応度を用いた動的分散遺伝的アルゴリズムについて述べる。

本誘導システムでは、計算負荷軽減のため複数の制約条件をペナルティ法を用いて単目的最適化へと変換している。このペナルティ法を用いて複数の制約条件および目的関数を一つの目的関数として扱う場合に使用する重み係数の役割は2つ存在すると考えられる。

一つ目は、各項目（制約条件の項目、目的関数の項目、軌道生成の場合は軌道終端位置における目標位置との誤差距離、誤差方位、飛行禁止区域飛行距離など）が有次元である場合、その値の成長具合を同じ土俵で比較および和算できるようにすることである。軌道生成の例を取り上げた場合、距離の単位であるメートルと方位の単位であるラジアンを単純に和算した場合、最適化においてはメートルがラジアンよりも取り得る値の範囲が大きいため、距離が優先的に最適化されやすくなってしまうことである。これを防ぐために、重み係数を用いて無次元化を行い、同じ成長具合の場合には、同じ値がとれるように重み係数を調整するのである。

二つ目は、各項目間の優先度を設定する目的である。この優先度が重み係数本来の目的であり、重み係数が大きいくほど、項目の変化に対する適応度への影響が大きくなるため、最適化においては重要視されるようになる。軌道生成を例に挙げた場合、目標位置における方位よりも目標位置に到達することを優先とする場合、到達位置誤差の重み係数を方位角誤差の重み係数よりも大きくすることで最適化では、到達位置誤差が優先的に満たされるように処理される。

以上のことから、ペナルティ法で用いている項目に関して、ファジィ集合の概念を取り入れることにより、項目値の無次元化と成長具合を定量的に扱うことが可能となる。そのため、ペナルティ法で用いている重み係数は、本来の目的である最適化の優先度として扱われるようになると考えられる。この手法について、軌道生成シミュレーションを行い、従来の軌道生成手法と比較し考察する。

3.2. ファジィ集合を用いた適応度計算

3.2.1. ファジィ集合

良いか悪いかの判断が曖昧である対象を定量化し、通常の集合と同じような演算を可能としたものをファジィ集合という[21]。たとえば、ある通学距離の長さ x について、A,B,C という 3 人の基準によって分けることを考える。歩くことを前提とした場合、通学距離が長いと感じる距離は 3 人とも別々の意見を持っていると思われるが、ファジィ集合を用いてこれらを定量化する場合、次のように数学的に表されるとする。

$$\mu_A(x) = 0.2$$

$$\mu_B(x) = 0.5$$

$$\mu_C(x) = 0.7$$

ここで、 $\mu(x)$ の数値が大きいほど距離が長いと感じるとする場合、A にとってこの通学距離は短いと感じている。同様に B にとってはそこそこの距離であり、C にとっては長いと感じている。もちろん、この通学距離は長さであるので、明確に数字で表すことが可能であるが、そこに曖昧な基準（この場合は 3 人の感覚）を追加し、その基準によってこの距離を表した場合に、定量的に評価できるようにしたものがファジィ集合である。このときの基準を表した関数 $\mu(x)$ をメンバシップ関数といい、通常 0 から 1 の間で定義される。

メンバシップ関数のイメージを図 3.2-1 に示す。ある評価対象について、目標値と最悪値を決定し、式(3.2-1) の様な形で定義することが可能である。ここで g_i^* は評価対象の目標値（理想値）であり、 g_i' は最悪値（希求値）である。

$$\mu_i(f_i) = \begin{cases} 1 & : (f_i \leq g_i^*) \\ 1 - \frac{f_i - g_i^*}{g_i' - g_i^*} & : (g_i^* < f_i \leq g_i') \\ 0 & : (f_i > g_i') \end{cases} \quad (3.2-1)$$

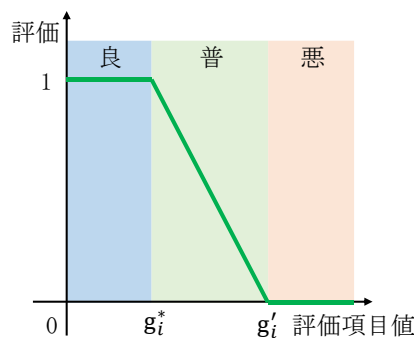


図 3.2-1 直角台形型メンバシップ関数

3.2.2.軌道生成におけるファジィ集合の適用

第2章で述べたように、DynDGAを用いた最適化の設定値である、適応度の重み係数は、これまで試行錯誤的に調整を行ってきた。この問題を解決するには、各評価指標を同じ次元で比較、積算できるようにする必要がある。

そこで、ファジィ集合を用いて、この評価指標をメンバシップ関数の評価値に変換し、各項目間の次元を取り払うことを考える。

現在適応度で用いられている評価指標は「到達地点誤差」「到達地点方位角誤差」「飛行禁止区域飛行距離」「飛行距離」の4種である。これらの項目についてそれぞれ目標値と最悪値を設定し、0から1の数値で表すようにする。これによって、設計者が設定した優先度と適応度の重み係数が比例的（または反比例的）に影響し合うようになると考えられる。

それぞれの目標値と最悪地の設定を以下の通りに定義する。

<到達地点誤差>

有翼ロケットが車輪で着陸することを考える際、実際の誘導システムでは、滑空誘導を本誘導システムで、そのあとの着陸フェーズは別の誘導システムで行うのが有効的であると考えられる。

そこで、目標地点から一定範囲内に有翼ロケットを滑空誘導できれば、問題なく着陸フェーズへ移行することが出来るとすると、その範囲は有翼ロケットの最小旋回半径に依存すると考えられる。そこで目標地点の高度において、最小旋回半径によって描かれる旋回円の直径を半径とする目標地点を中心とした円内を目標と定義した。

GAによって最適化を行う場合、解の評価を的確に行う必要がある。特に、良好な解、不良な解の2種類に分類する場合、不良な解の中でもランク付けが行える方が最終的に最適解を得やすくなる。つまり、最悪値は取り得る最大誤差とすべきである。よって、到達地点誤差の最悪値は、有翼ロケットの飛行可能距離を基準に、初期地点から目標地点とは反対方向へ初期方位角と関係なく直線で飛行した場合の軌道終端位置と目標地点との距離とした。

図 3.2-2 にそのイメージを示す。

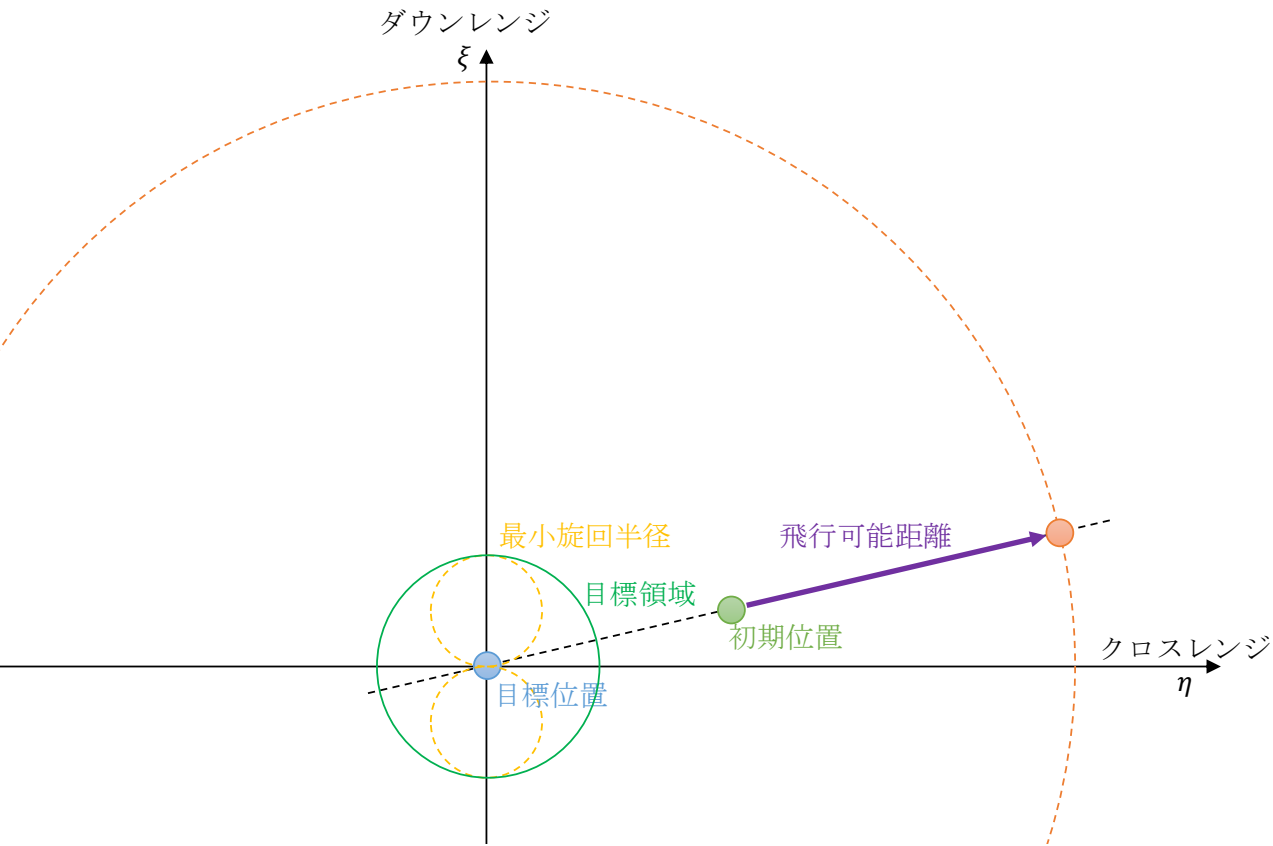


図 3.2-2 到達地点誤差のファジィ化イメージ

最小旋回半径 R_{\min} は以下の式から求められる。

$$R_{\min} = \frac{V^2}{g \tan(\phi_{\max})} \quad (3.2-2)$$

なお, g は重力加速度, V は旋回速度, ϕ_{\max} は最大バンク角で本論文の場合 $\frac{\pi}{3}$ [rad]である。旋回速度は次の式から得られる。

$$V = \sqrt{\frac{2W}{\rho S_w C_L}} \quad (3.2-3)$$

W は機体重量であり, ρ はその高度における空気密度, C_L は揚力係数である。この場合, 揚力が大きいほど旋回半径は小さくなるため, C_L は取り得る最大値を代入する。

飛行可能距離 L_{cap} は高度に依存する。揚抗比最大とした場合に, 直線で飛行すると飛行距離は最も伸びる。本来は飛行速度に依存するが, 簡易化のため, 特定飛行速度における揚抗比最大となる迎角で飛行した場合の飛行距離を飛行可能距離と定義する。

$$L_{\text{cap}} = \max\left(\frac{L}{D}\right) \times (\text{Alt}_I - \text{Alt}_T) \quad (3.2-4)$$

ここで $\max\left(\frac{L}{D}\right)$ は最大揚抗比，Altは高度を表し，添え字はIが初期位置，Tが目標位置とする．

<到達地点方位角誤差>

到達地点における方位角と目標地点における方位角の誤差を到達地点方位角誤差と呼ぶ．この到達地点方位角誤差のメンバシップ関数イメージを図 3.2-2 に示す．

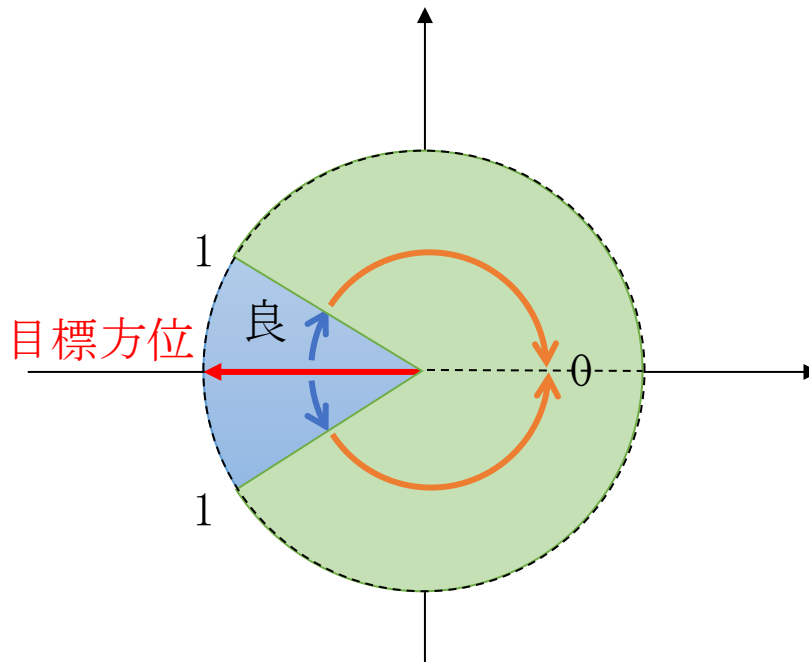


図 3.2-3 到達地点方位角誤差メンバシップ関数イメージ

<飛行禁止区域飛行距離>

飛行禁止区域を飛行する距離は無いのが理想である．故に，目標値は 0 に設定する．

一方で，最悪値は，飛行禁止区域飛行距離を飛行できる距離に設定するべきであるが，その場合，飛行禁止区域内を旋回等によってずっと飛行し続ける可能性があるため，式(3.2-4) の飛行可能距離 L_{cap} を最悪値とすることも可能であるが，そのような個体はそもそも最適化で使用するべきでは無いと考える．故に，本論文のように，長方形で設定されている飛行禁止区域の場合，その長方形の対角線長さを最悪値と設定する．

飛行禁止区域のメンバシップ関数イメージを図 3.2-4 に示す.

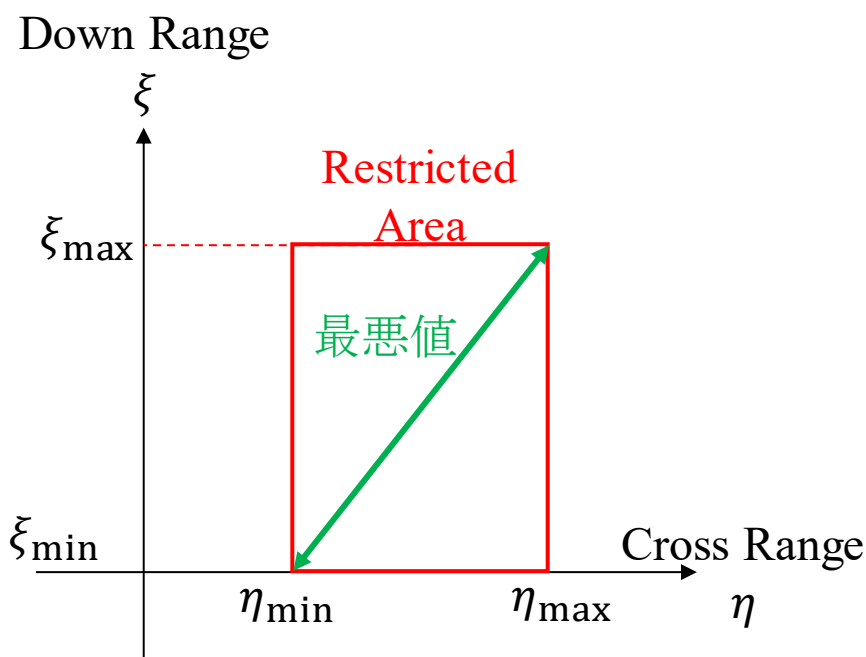


図 3.2-4 飛行禁止区域のメンバシップ関数

< 飛行距離 >

飛行距離の最悪値は、式(3.2-4) の飛行可能距離 L_{cap} とした.

目標値については、できる限り飛行距離は短いものが良いため、0 とするのが一般的である. しかし、目標値を 0 とした場合、墜落するような軌道が、まっすぐ目標地点へと飛行する軌道よりも優れているようになってしまう. そこで、飛行距離の目標値は初期位置と目標位置における方位角は考慮外とし、単純な初期位置と目標位置の距離の 0.9 倍とした. 10%の余裕を持たせることにより、最適化の際に、多少墜落してしまっている軌道も評価の対象とするためである. 現在は、試行的にシミュレーションへ適応するため、図 3.2-1 に示す様な直角台形型のメンバシップ関数を用いているが、今回のような場合には、図 3.2-5 に示すような凸台形型を用いて、一定範囲内の評価値が優良であるようにメンバシップ関数を設定することも試してみるべきである.

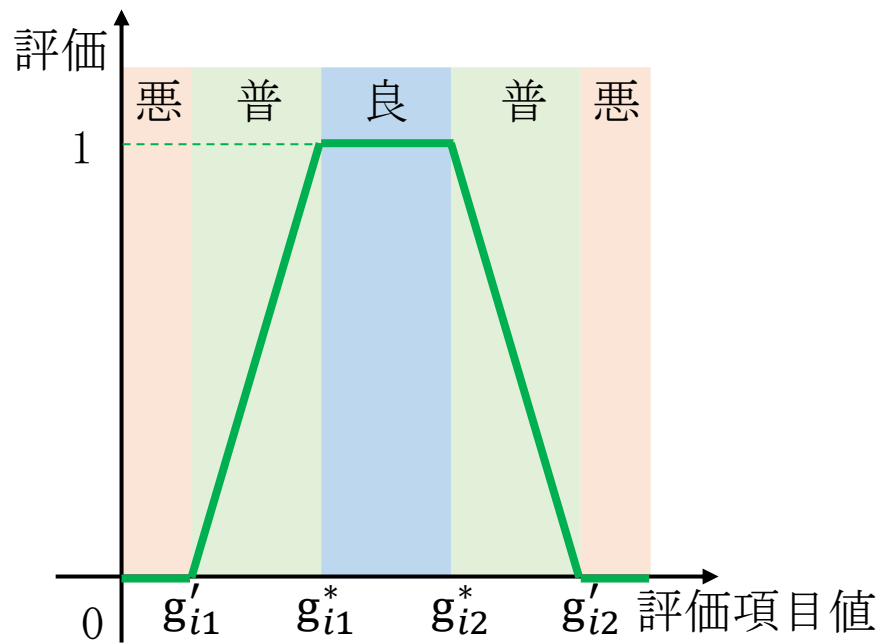


図 3.2-5 凸台形型メンバーシップ関数

3.2.3. ファジィ集合を用いた適応度

3.2.2 より，軌道最適化において使用する適応度を式のように定義する．

$$\text{fit}_{\text{fuzzy}} = 100 \sum_{i=1}^4 (w_i \mu_i) \quad (3.2-5)$$

ここで， $\text{fit}_{\text{fuzzy}}$ が適応度値， w_i は重み係数， μ_i はメンバーシップ関数である．この重み係数は，評価指標の優先度が高い項目ほど大きな数字となる．

3.2.4. ファジィ集合を用いた適応度による軌道生成シミュレーション

ファジィ集合を用いた適応度による軌道生成シミュレーションを行った．

飛行領域は図 2.3-18 と同様であり，計算条件も表 2.3-1 と同様である．

ファジィ集合のメンバーシップ関数における目標値と最悪値は表 3.2-1 の通りである．また，適応度の重み係数は表 3.2-2 の通りである．

表 3.2-1 メンバシップ関数の設定

	目標値 g_i^*	最悪値 g_i'
到達地点誤差	180[m]	46,138[m]
到達地点方位角誤差	1[deg.]	180[deg.]
飛行禁止区域飛行距離	0 [m]	2236[m]
飛行距離	10,036[m]	46,138[m]

表 3.2-2 ファジィ集合を用いた手法のシミュレーションにおける重み係数

到達地点誤差	到達地点方位角誤差	飛行禁止区域飛行距離	飛行距離
100	7	10	1

シミュレーション結果を図 3.2-6 から図 3.2-13 に示す。

目標地点へ到達できる軌道が得られた。また、解の成長率も調整前に比べて良くなっている。一方で、表 3.2-2 のような重み係数を設定するまでに何度かシミュレーションを行う必要性があった。迎角プロファイルに関しても、異常に振動している結果が得られている。迎角の振動に関しては、機体の応答性を考慮せずに最適化を行っているからであり、今後最適化を行う際の各制御入力 of 周波数遺伝子に関して、機体の応答性や固有振動数などから制限を設ける必要があると考えられる。

本結果では見られなかったが、別条件でシミュレーションを行った場合、設計者が設計した優先度に沿った評価項目の成長が行えていないパターンや同じ優先度に設定された評価項目間で成長率に大きな差異が見られる（図 3.2-14）。その理由として、ファジィ集合へ評価指標を変換することにより、解の成長具合を数値で判断できるようになると同時に、重み付け和による次元は同一化できたものの、評価指標の「成長しやすさ」という面においては、各項目に差異が生じてしまったためだと考えられる。解の成長しやすさまでを考慮することはファジィ集合では行えないと考えられるため、別の手法によってこの課題を解決する必要がある。

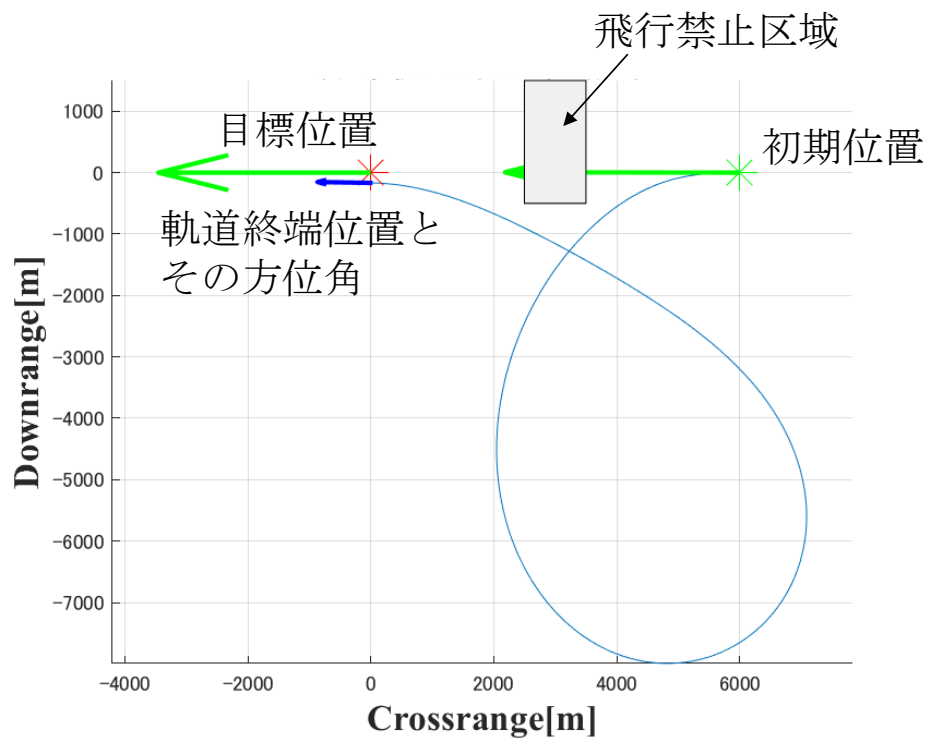


図 3.2-6 ファジィ集合を用いた誘導軌道最適化結果(最適軌道)

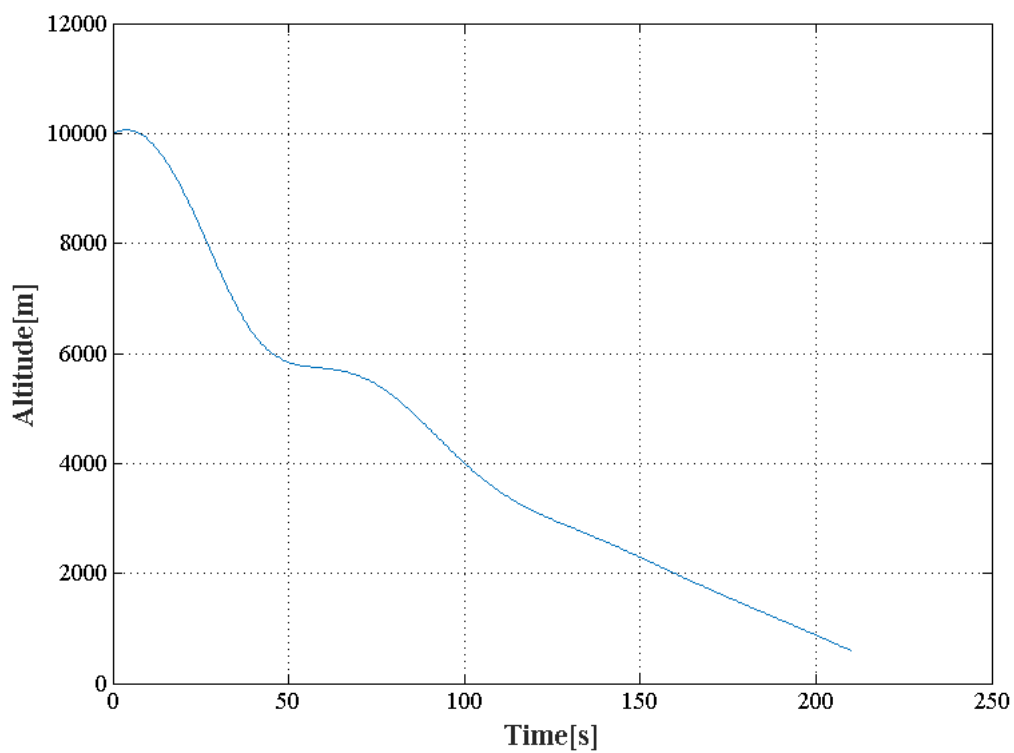


図 3.2-7 ファジィ集合を用いた誘導軌道最適化結果(高度プロファイル)

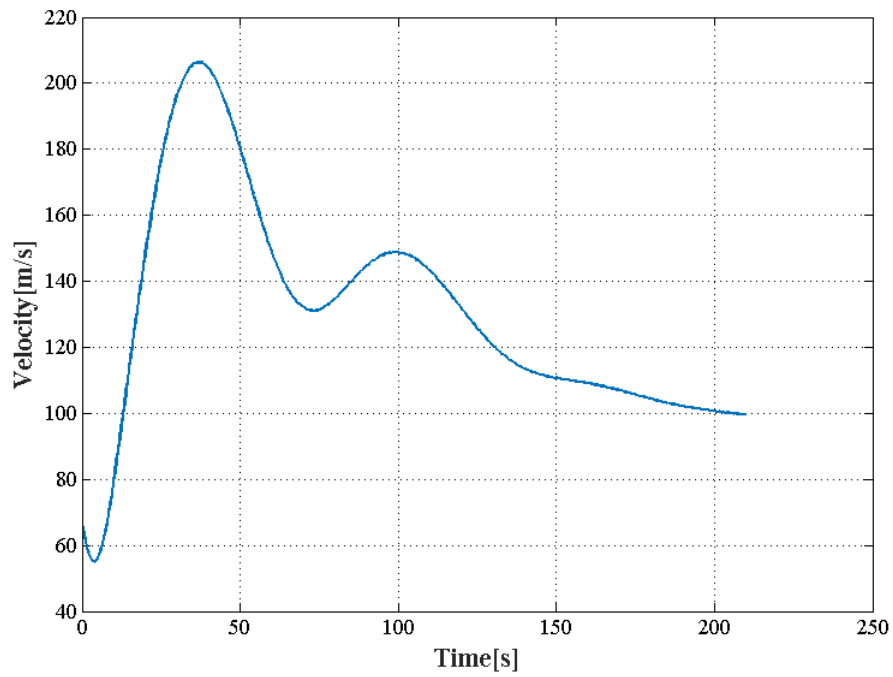


図 3.2-8 ファジィ集合を用いた誘導軌道最適化結果(速度プロファイル)

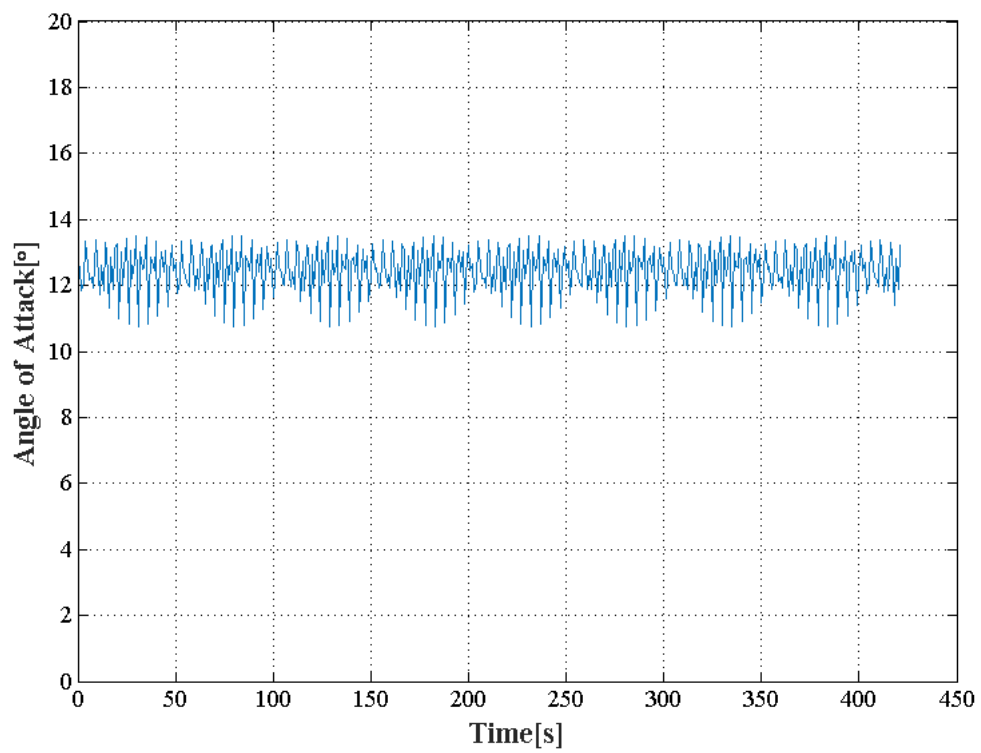


図 3.2-9 ファジィ集合を用いた誘導軌道最適化結果(迎角プロファイル)

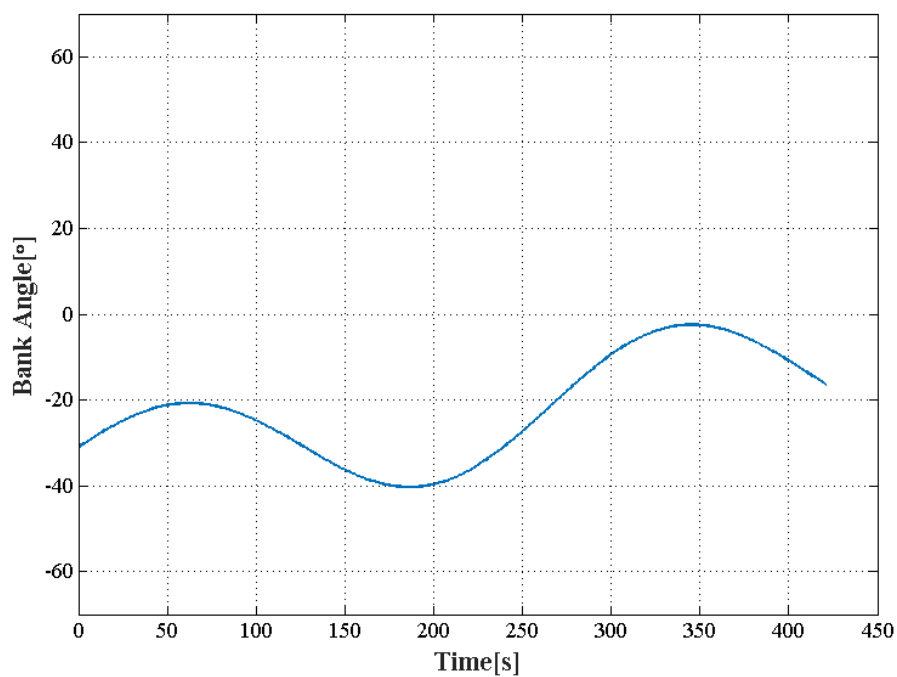


図 3.2-10 ファジィ集合を用いた誘導軌道最適化結果(バンク角プロファイル)

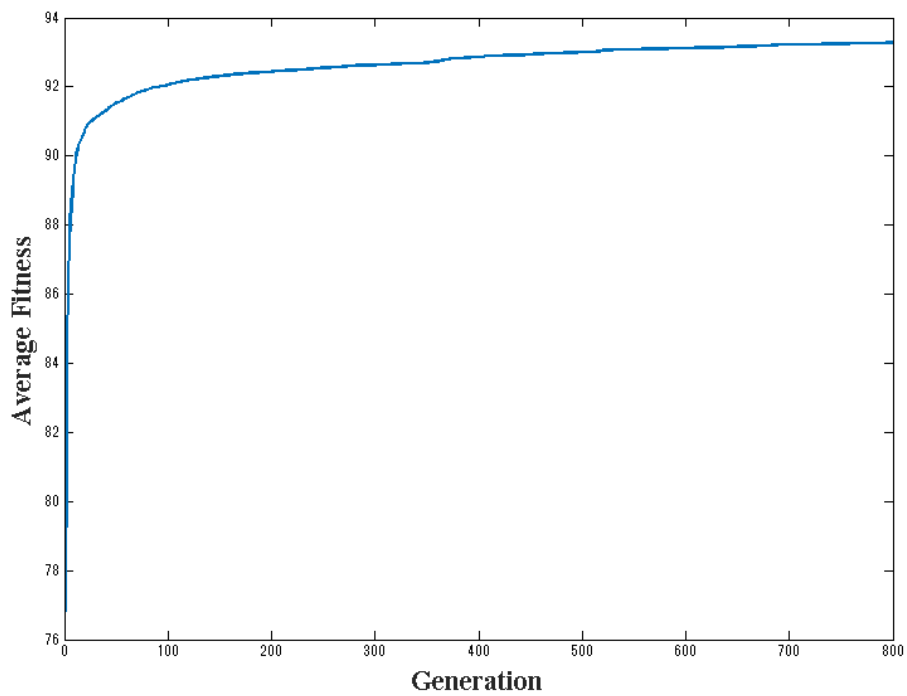


図 3.2-11 ファジィ集合を用いた誘導軌道最適化結果(平均適応度)

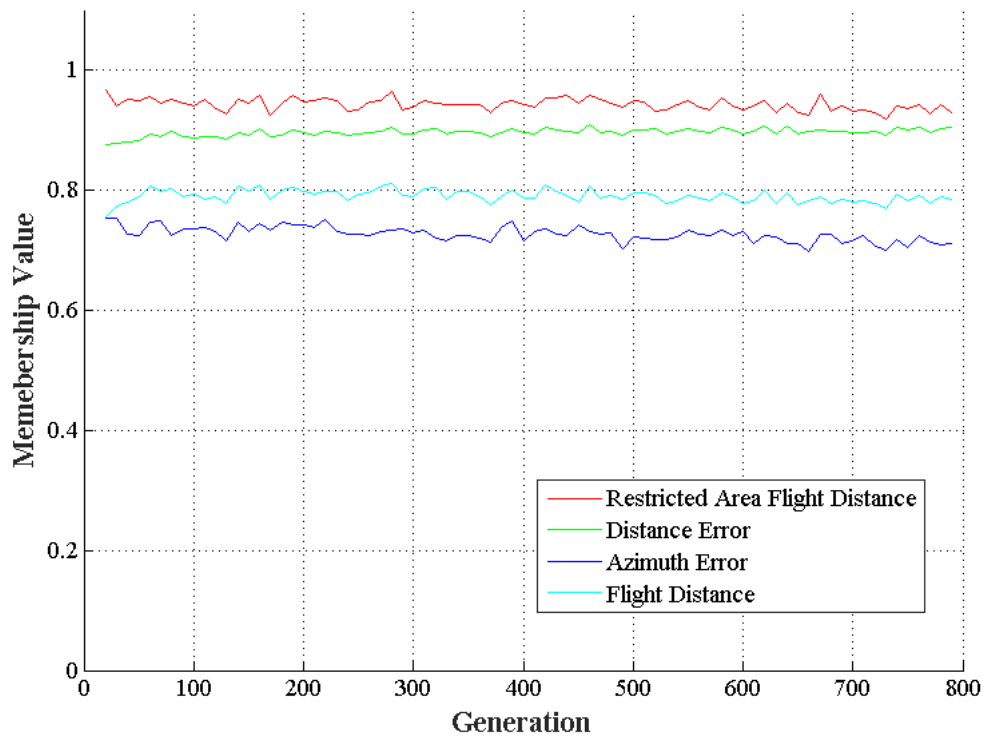


図 3.2-12 ファジィ集合を用いた誘導軌道最適化結果(メンバシップ関数値)

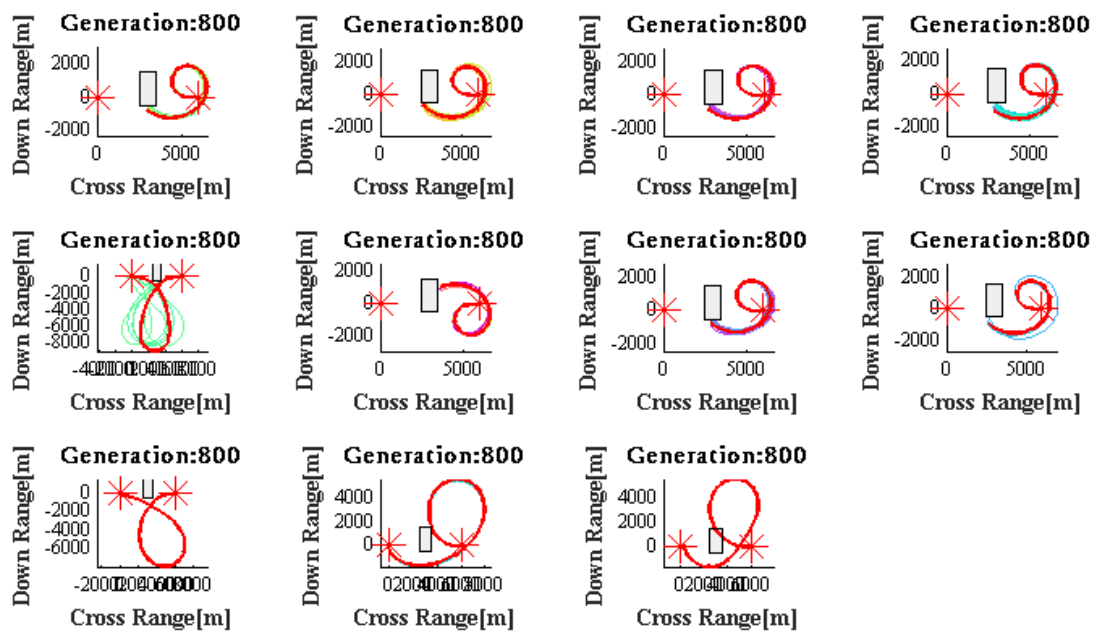
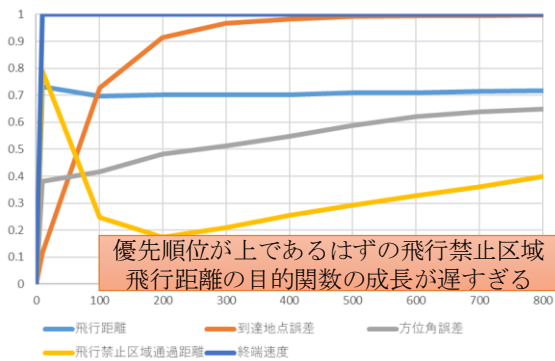
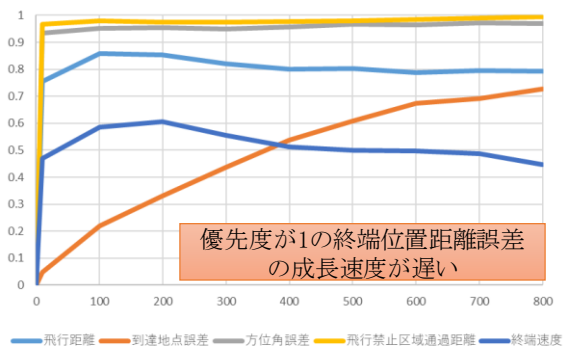


図 3.2-13 ファジィ集合を用いた誘導軌道最適化結果(最終世代の軌道)



優先順位	項目
1	終端速度
2	終端位置距離誤差
3	飛行禁止区域飛行距離
4	終端位置方位角誤差
5	飛行距離



優先順位	項目
1	終端位置距離誤差
1	終端位置方位角誤差
1	飛行禁止区域飛行距離
2	飛行距離
3	終端速度

図 3.2-14 優先順位が反転するパターン

3.3.階層的手法を用いた適応度計算

3.3.1.評価指標の優先度と階層的手法

前節で示したように、適応度にファジィ集合の概念を組み込んだ手法では、次元の削減に成功したものの、評価指標の「成長しやすさ」までは吸収することが出来ず、結果としてペナルティ法の重み係数において「成長しやすさ」と「優先度」を設定する必要があった。この「成長しやすさ」は飛行領域の設定や機体モデル、初期位置と目標位置の関係などで変化するため、試行錯誤的に重み係数を調整する必要がある。

そこで、各評価関数に目標値を設定したうえで、それらに優先順位をもうけることで、ill-defined な軌道最適化問題に対しても、設計者の要求を反映した意味のある準最適解が得られる手法[22]を参考に、ファジィ集合を用いた階層的適応度計算手法を提案する。

この手法は、予め設計者が設定した評価項目の優先度に従って最適化を行う手法で、優先度（階層）の高い評価項目の値を超えない範囲で、その項目より低い優先度が設定されている評価項目は最適化が行われるという手法である。

式(3.3-1)で示す様に、評価項目の優先度は同じ優先度を設定することが可能である。

$$\text{fit}_{\text{clstr}} = \frac{100}{\sum_{i=1}^L P_i} \left(\frac{P_1}{\sum_{i=1}^{K_1} w_{1i}} \sum_{i=1}^{K_1} w_{1i} \mu_{1i} + \cdots + \frac{P_L}{\sum_{i=1}^{K_L} w_{Li}} \sum_{i=1}^{K_L} w_{Li} \mu_{Li} \right) \quad (3.3-1)$$

ここで、Lは階層の数、Kはその階層における評価項目の数、Pは先制優先因子、 w_{ij} は階層 i における j 番目の評価項目の重み係数である。

3.3.2.先制優先因子

先制優先因子は、その階層の優先度を決定する値であり、 $P_i \gg P_{i-1}$ の関係を持つ。一番低い階層の先制優先因子を 1 とすると、その一つ上の階層では 10 倍以上の数値が設定されるのが一般的である。

本誘導システムにおいては、以下の式(3.3-2)により、先制優先因子を決定している。

$$P_i = 10^{L-i} \quad (3.3-2)$$

ここで、Lは階層の数を表す。

3.3.3. ファジィ集合を用いた階層的適応度計算手法による軌道生成

シミュレーション

ファジィ集合を用いた階層的適応度計算手法による軌道生成シミュレーションを行った。飛行領域および機体条件等は前章と同様である。また、同一階層には一つの評価項目のみを設定しているため、重み係数は1である。

図 3.3-1 から図 3.3-7 にシミュレーション結果を示す。

最適軌道が目標地点へ到達できていることが解る。また、ファジィ集合のみを適応したシミュレーションと比較して、方位角誤差の項目と飛行距離の項目について、設計者が指定した優先度に従って解が成長していることが確認できた。

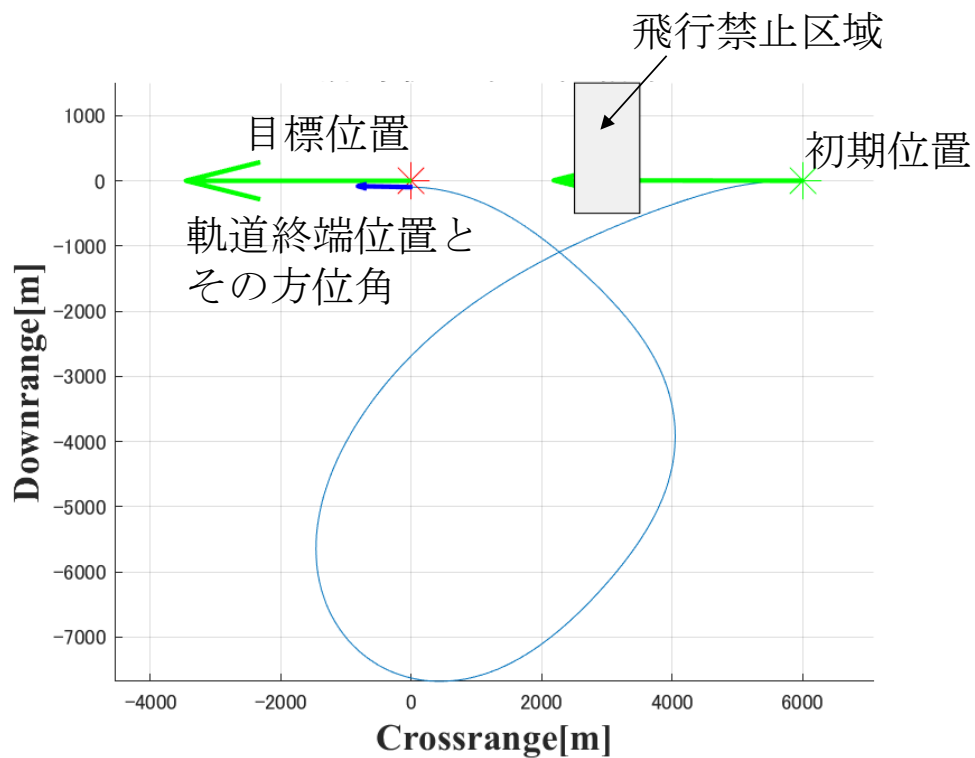


図 3.3-1 階層的的手法による軌道最適化結果(最適軌道)

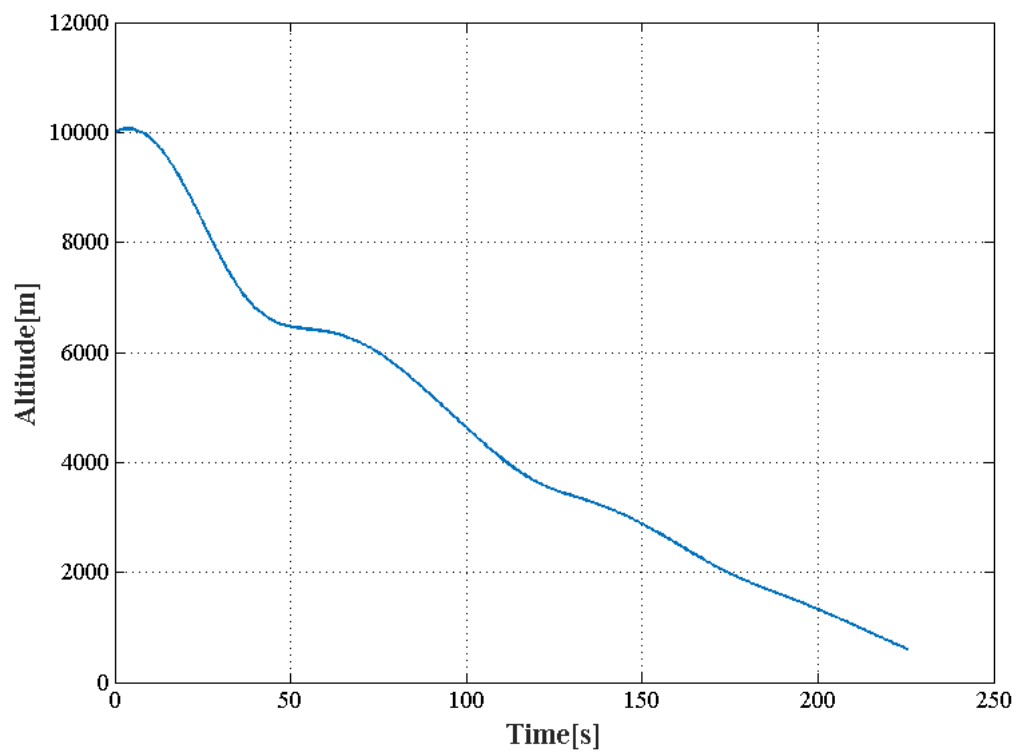


図 3.3-2 階層的手法による軌道最適化結果(高度プロファイル)

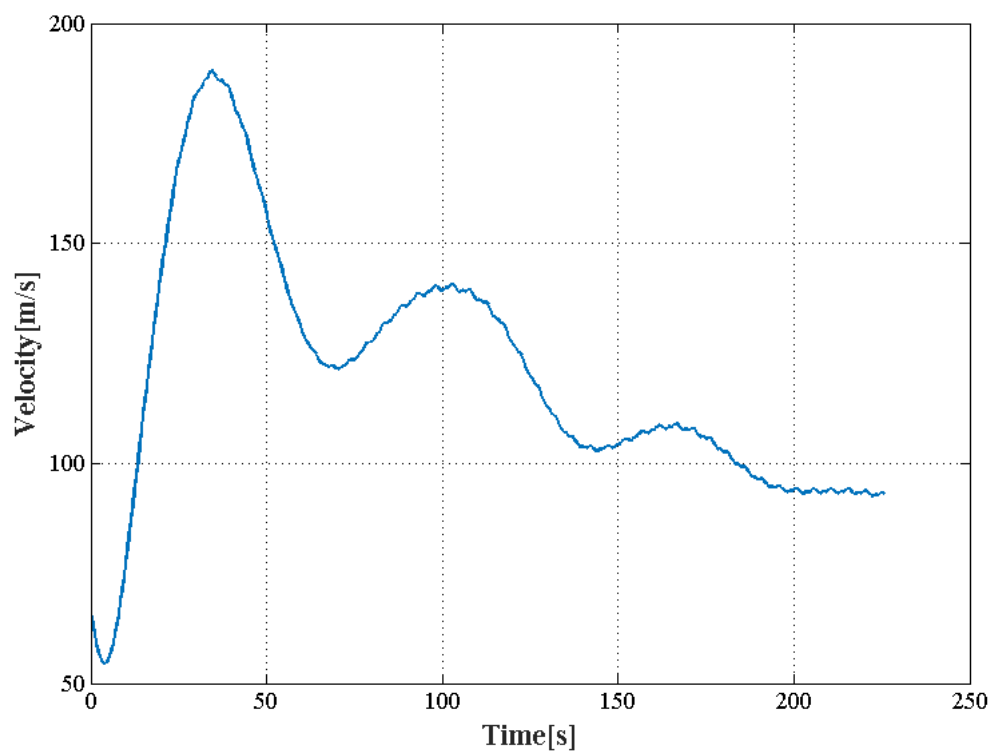


図 3.3-3 階層的手法による軌道最適化結果(速度プロファイル)

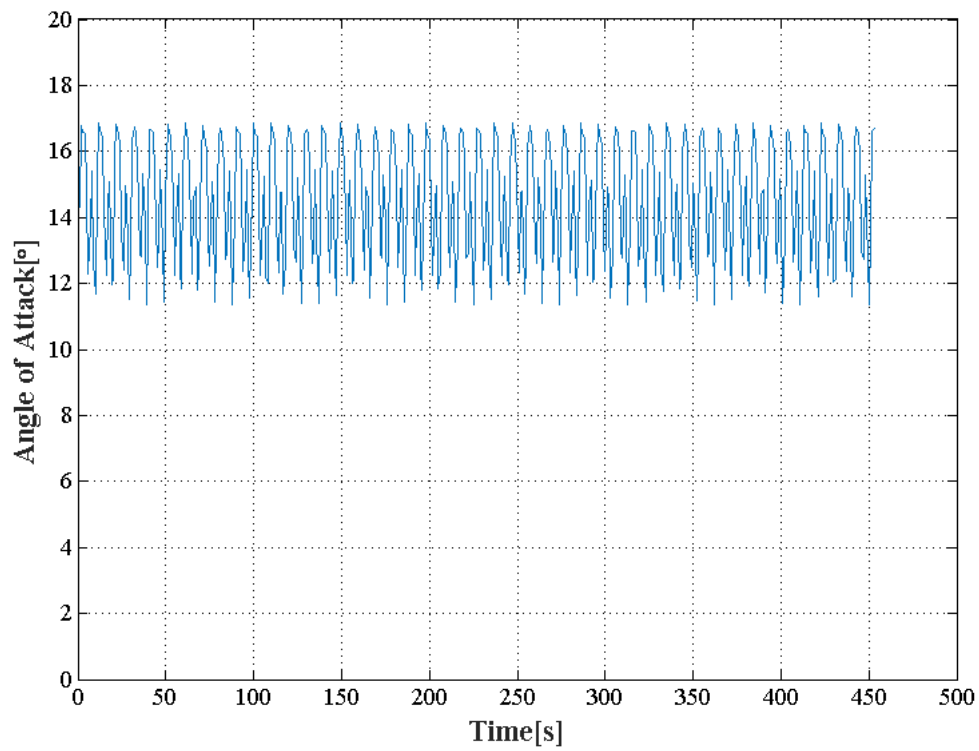


図 3.3-4 階層的手法による軌道最適化結果(迎角プロファイル)

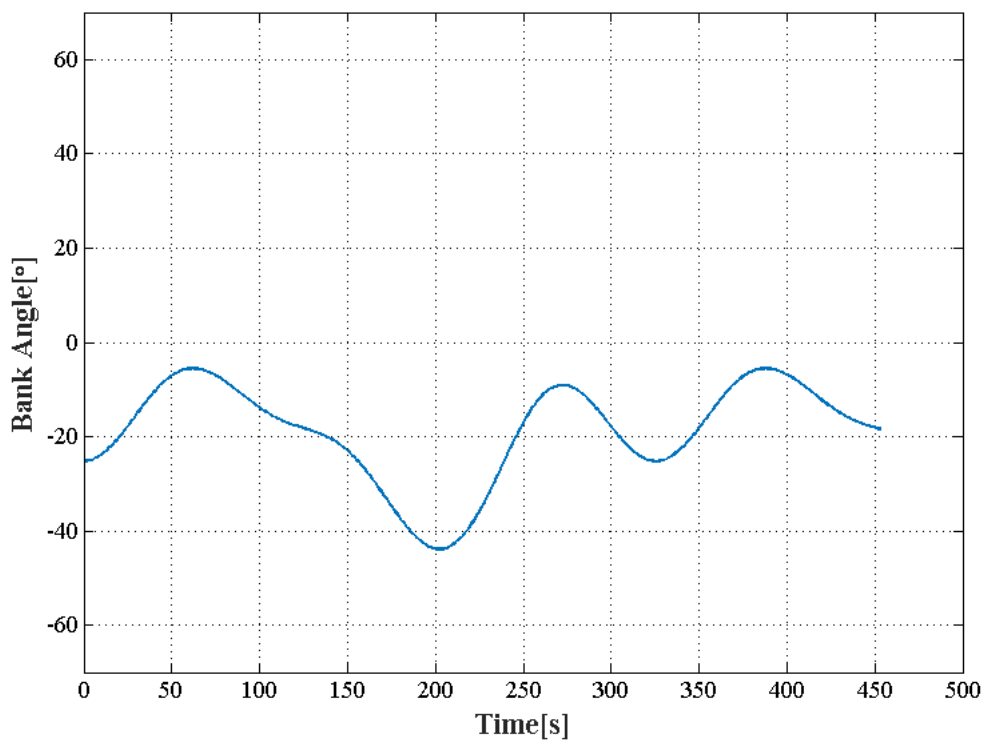


図 3.3-5 階層的手法による軌道最適化結果(バンク角プロファイル)

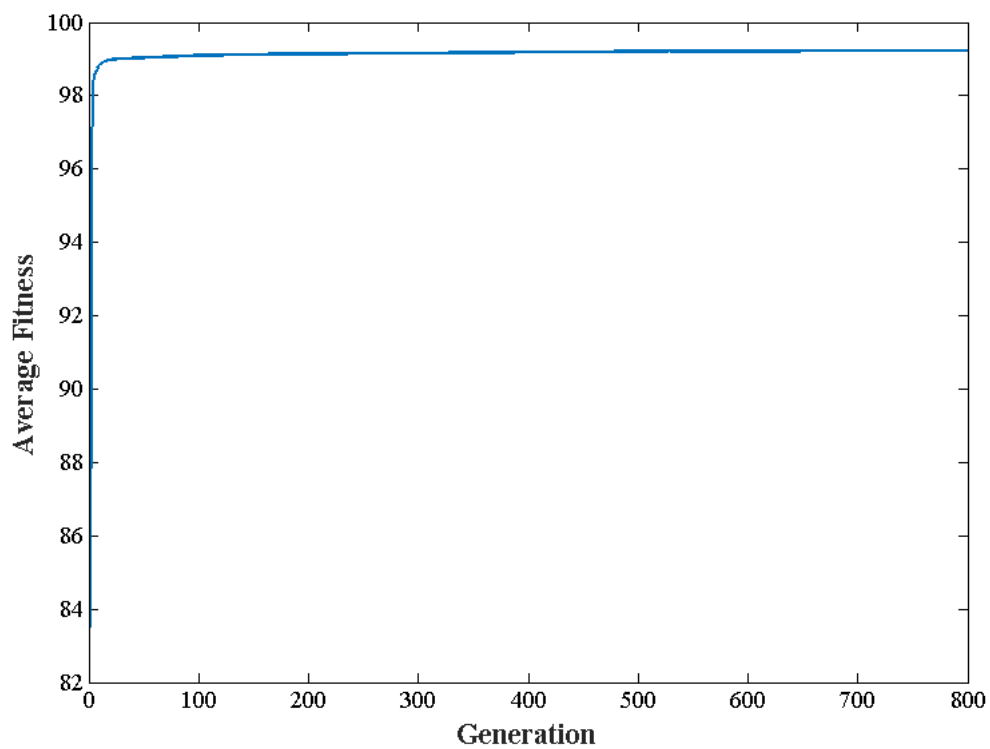


図 3.3-6 階層的手法による軌道最適化結果(平均適応度)

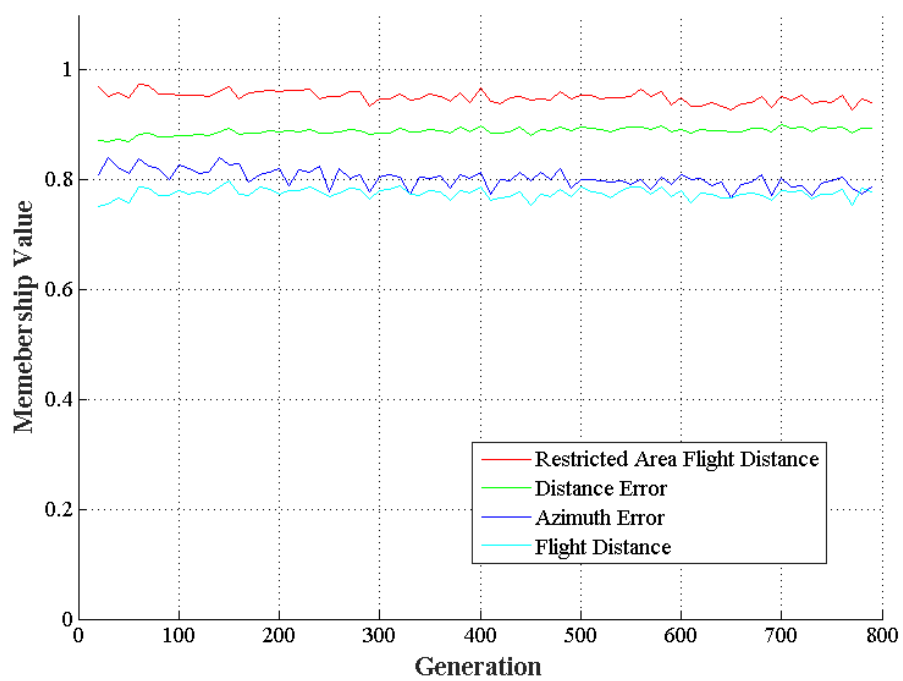


図 3.3-7 階層的手法による軌道最適化結果(メンバーシップ関数値)

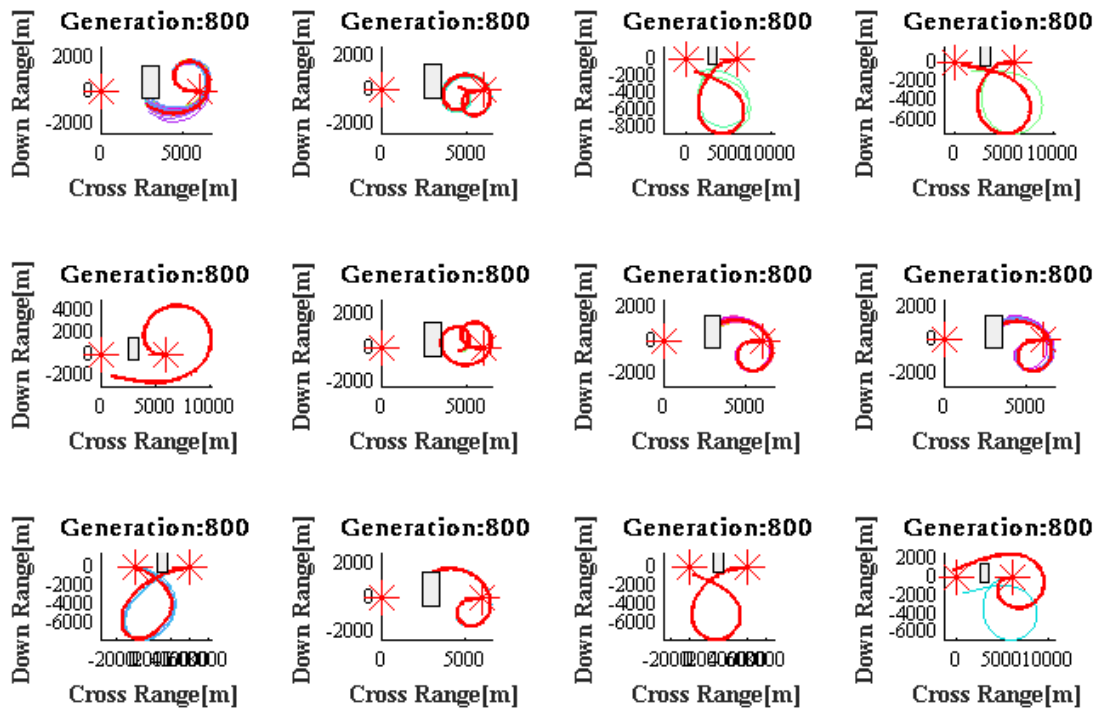


図 3.3-8 階層的手法による軌道最適化結果(最終世代の軌道)

3.4. 適応度に使用する係数の自動調整手法

式(3.3-1)に示す様に，同一階層内における項目の重み係数は，未だに試行錯誤的に設定する必要がある．また，先制優先因子の数値の設定にも不明瞭な点が多く残っている．そこで，さらなる汎用性向上のため，これらの係数の設定を自動で行う手法を提案する．

自動調整を行うに当たり，各係数の役割を以下のように定義した．

先制優先因子：階層間の優先度に従った解の成長を促すための係数

重み係数：同一階層内における各評価項目の成長率を同一にする係数

これにより，重み係数を「成長率調整係数」と呼称する．

2 係数の自動調整方針を以下のように纏める．

<先制優先因子>

各クラスタにおいて，同じ階層の評価項目の平均値を取得し，その平均値が一つ下の階層を下回っていたら，自分以上の階層の先制優先因子である 2^n の次数を加算する．

このとき，自分以上の階層の先制優先因子すべての次数を加算するのは，自身と他の階層との相対距離を保つためである．

<成長率調整係数>

各クラスタにおいて，同一階層内の評価項目毎に平均値を取得し，その中で最も高い平均値の項目を基準に，その他の項目との値の差が μ_{th} 以上の場合，成長率調整係数である 2^n の次数を加算する．なお， μ_{th} は評価項目の誤差範囲であり，0 でも可能であるが，各評価項目の値が最適化途中で同一になることはほぼ無いため，0.05 と設定する．

このシミュレーション結果を図 3.4-1 から図 3.4-7 に示す．なお，自動調整の間隔は 10 世代毎とする．

自動調整により，評価項目の値が優先順位に則さない形で成長すると，係数が調整され，再び優先順位の則するようになる様子が見られた．適応度の成長率は圧倒的に早くなったが，優先順位に即する形で成長し続けることが無く，課題が残った結果となっている．

また，適応度のプロファイルより，係数が調整された際に，平均適応度が低下してしまうことがある．アルゴリズム上，仕方の無いことではあるが頻繁に変更

した場合、遺伝子処理に影響を与え、解の成長が悪くなってしまう可能性がある。現に、本結果では通常ファジー集合を用いた階層的技法の先制優先因子を固定した場合に比べ、解の成長率と解軌道の多様性に劣っていることが解る。

以上より、本手法は、係数の予測が全くつかない状況下において、一定の効力を発揮するものの、必要とされる最適化計算の世代数は通常よりも多く必要である可能性がある。実飛行環境下で使用する場合は、先制優先因子を固定とする法が、現状では良いと考える。一方で、本シミュレーションでは、成長率調整係数について、本手法が有効であるかの確認が出来ていないため、今後検討していく必要がある。

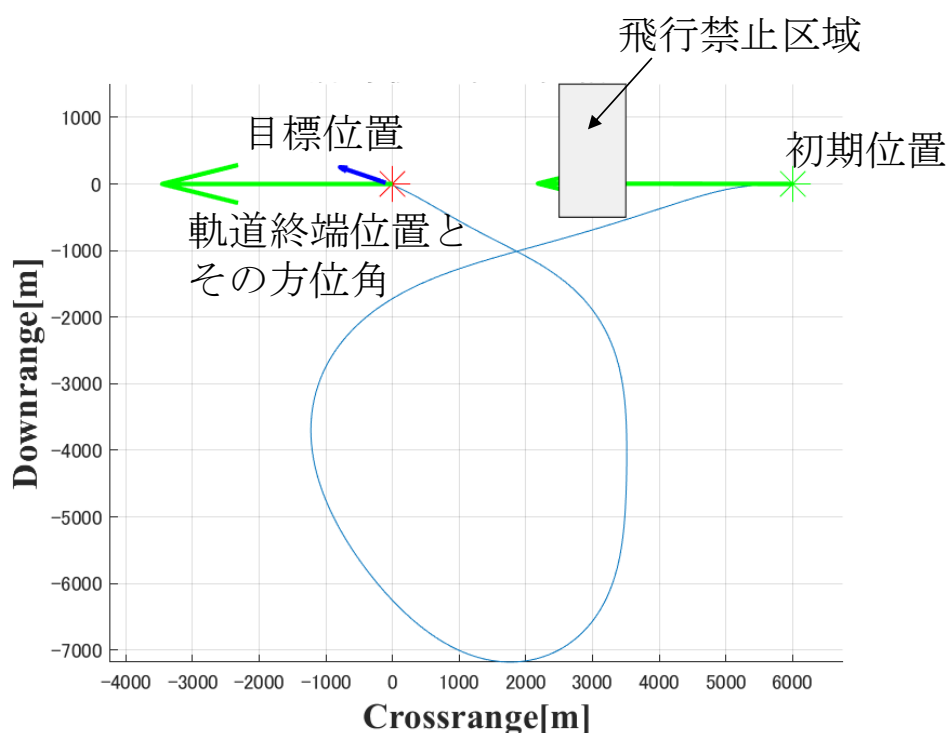


図 3.4-1 係数自動調整手法による軌道最適化結果(最適軌道)

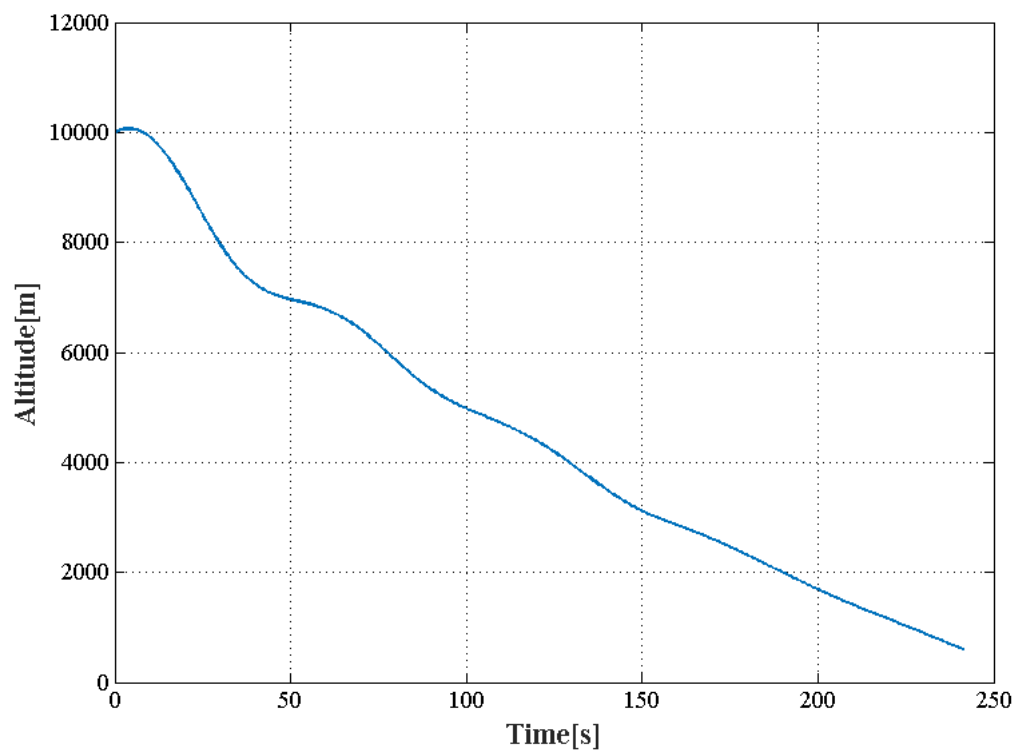


図 3.4-2 係数自動調整手法による軌道最適化結果(高度プロファイル)

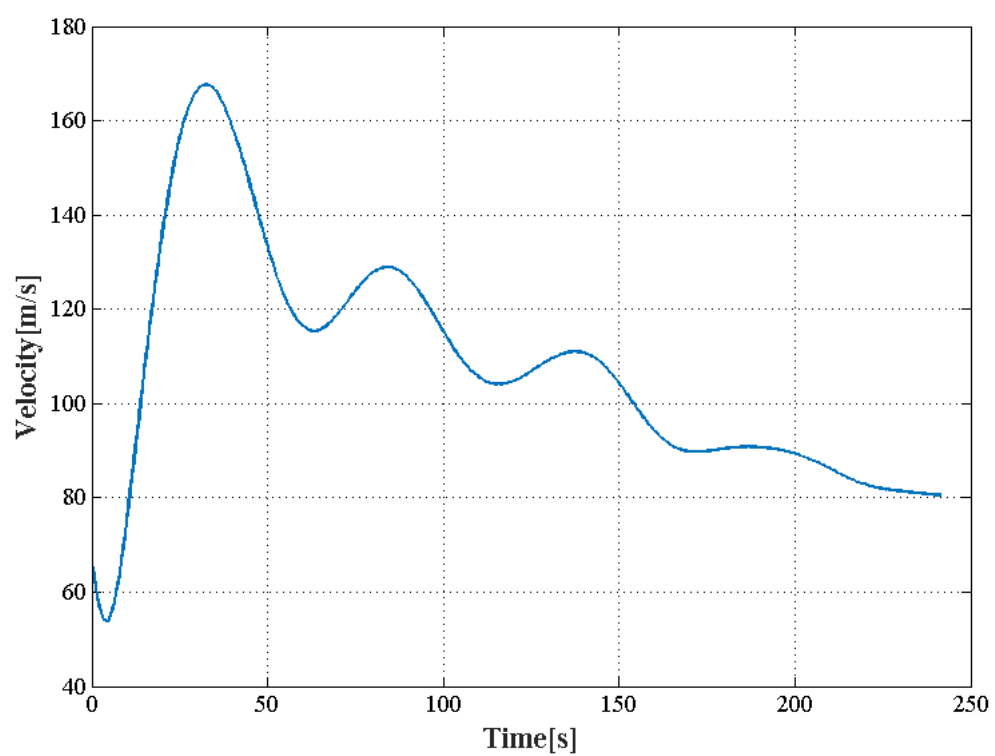


図 3.4-3 係数自動調整手法による軌道最適化結果(速度プロファイル)

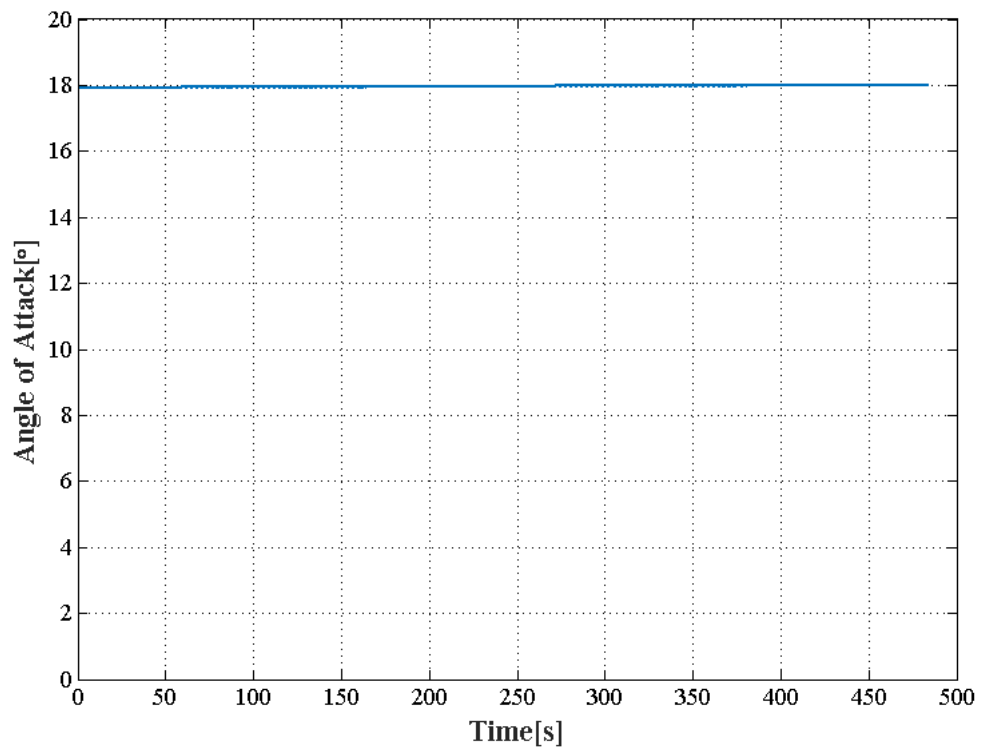


図 3.4-4 係数自動調整手法による軌道最適化結果(迎角プロファイル)

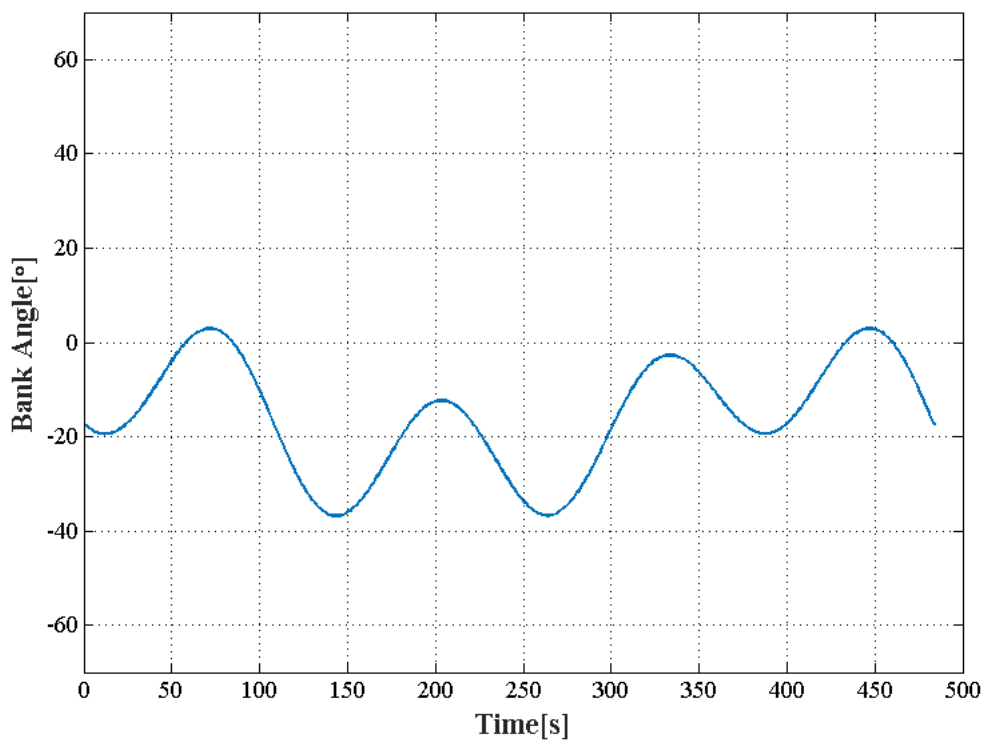


図 3.4-5 係数自動調整手法による軌道最適化結果(バンク角プロファイル)

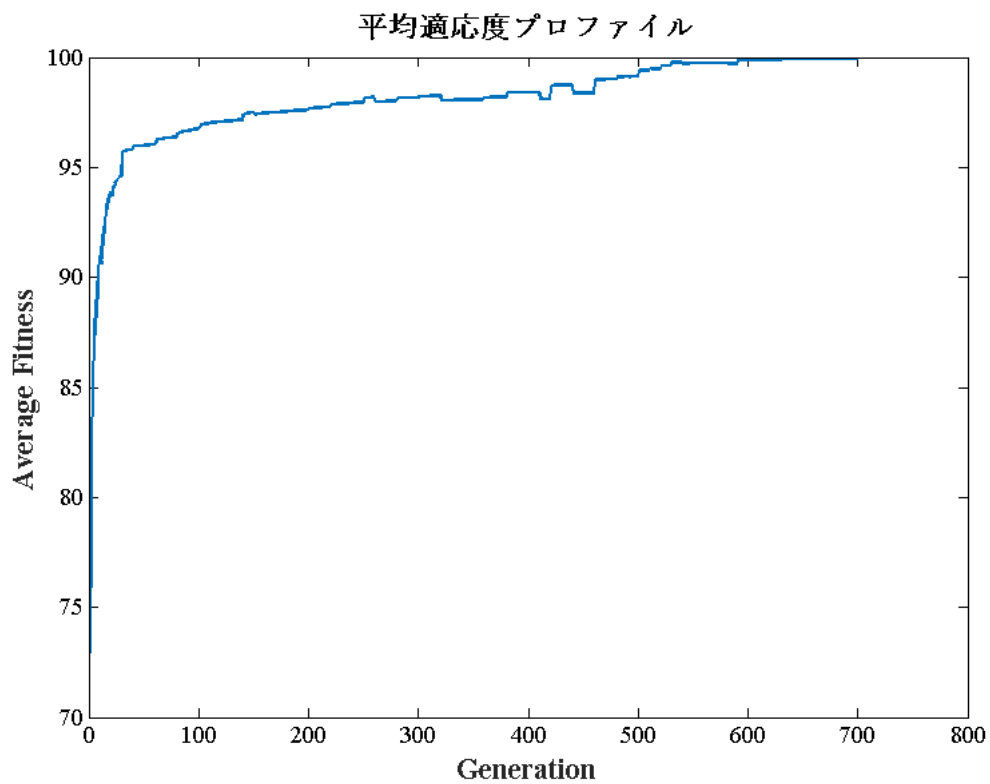


図 3.4-6 係数自動調整手法による軌道最適化結果(平均適応度)

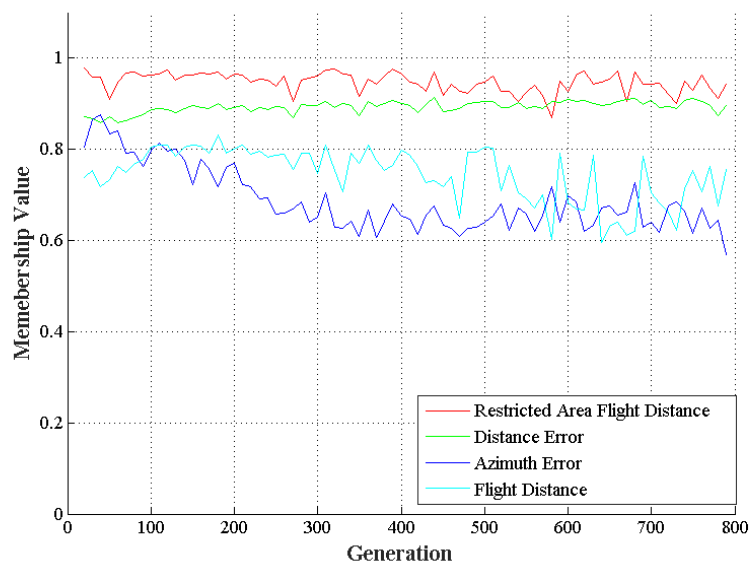


図 3.4-7 係数自動調整手法による軌道最適化結果(メンバシップ関数値)

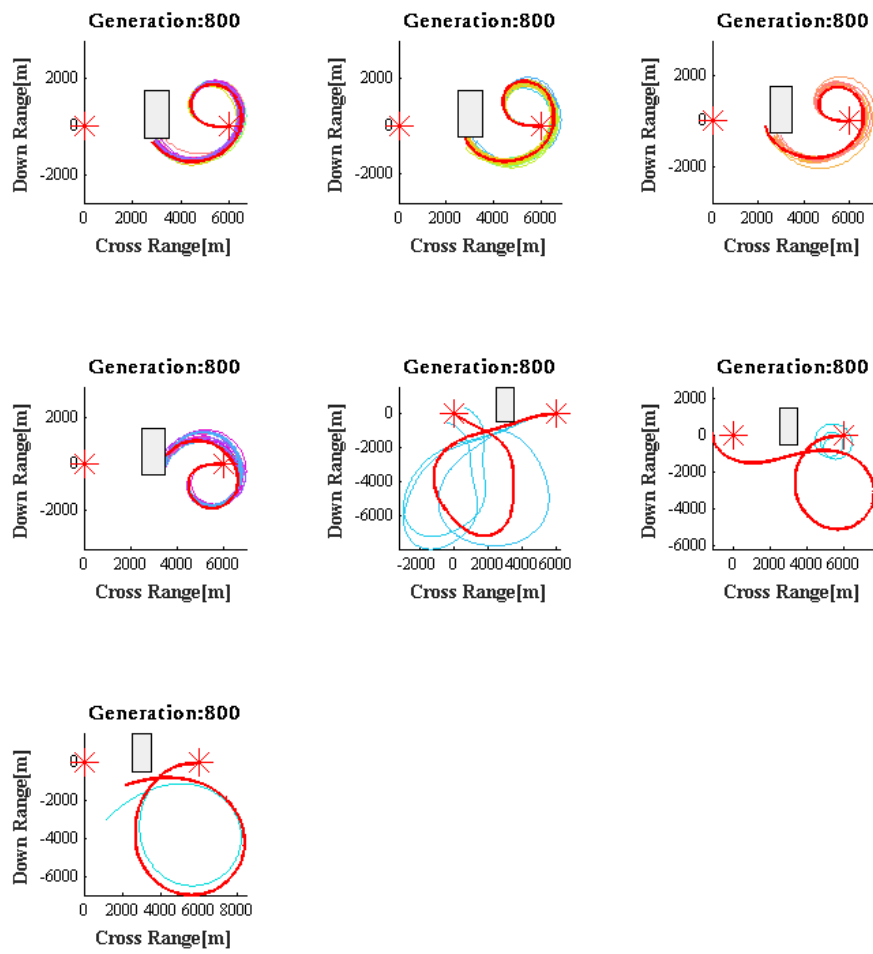


図 3.4-8 係数自動調整手法による軌道最適化結果(最終世代の軌道)

3.5. おわりに

第 3 章では、ファジィ集合を用いた階層的適応度計算を用いた軌道生成手法の提案を行った。この手法によって、設計者が予め設定しておく必要のある設定値の数を削減することに成功し、汎用性が向上した。

ファジィ集合を用いることにより、解の成長具合を数値的に評価できるようになったため、実用化の面においても本誘導システムによって生成された誘導軌道を滑空飛行に使用するかどうかの判断がコンピュータで行えるようになった。

また、ファジィ集合のみだけでは吸収することが出来なかった「評価項目の成長のしやすさ」についても、階層的手法を取り入れることにより、設計者が設定した評価項目の優先度に沿った解の探索を行うことが出来るようになり、最終的に解の探索性能が向上した。

最終的に、これらの係数を自動で調整するアルゴリズムを取り入れることにより、設計者が試行錯誤的に設定する係数が削減され、設計および最適化計算にかかる時間を短縮することの可能性を提示できたものの、先制優先因子については固定とすべきことが解った。一方で成長率調整係数については、本シミュレーションでは確認できなかったため、今後確認する必要がある。

迎角プロファイルにおいて、一部の軌道生成結果に制御入力が振動してしまう結果が見られた。この対策として、機体の応答性と固有振動数などから最適化変数の振動数の項について制限を設ける必要があることがわかった。

第4章 DynDGA を用いたリアルタイム最適軌道生成システム実用化の検討

4.1. はじめに

第 3 章では、ファジィ集合を用いた階層的適応度計算手法により、設計者が調整する必要のあった設定値の一つ「適応度の重み係数」について、ファジィ集合による次元の削減と階層的手法による優先度に従った最適化が行えるようになった。また、評価項目値の成長度合いによって自動的にこれらのパラメータを調整するアルゴリズムの提案を行った、これにより、軌道生成システムの汎用性向上へと繋がった。

本章では、第 2 章で挙げた課題の一つ、計算コストに関して、誘導システムを実飛行で使用するに当たっての検討を述べる。従来研究において、DynDGA ではない通常の GA を用いた誘導軌道生成システムを飛行実証するに当たり、最も計算コストの高い処理である各個体の適応度計算部を並列化することと、ある程度良好な初期個体を生成し、そこから最適化を進めることにより、少ない個体数と世代数で優良な解を 1 秒以内に得ることが出来た。この手法を元に、DynDGA を用いた誘導軌道生成システムを構成することを考えた場合、必要とされる個体数と世代数は GA の数倍となるため、高価な計算機が複数必要となると予想される。また、円と直線を用いた定常滑空近似(後記の図 4.2-8 参考)による初期個体生成手法では、一部の条件下において解が算出できないことが解った。

そこで、安価なコンピュータにおいても、DynDGA を用いた誘導軌道生成システムにより、飛行中にリアルタイムで誘導軌道生成を行うため、最適化処理を続けたまま決められた誘導軌道更新周期においてその時の最適軌道を用いて飛行を行う手法を提案する。乱数を用いて最適化を行う DynDGA では、最適化が終了するタイミングが一樣に決定することが出来ない点からも、本手法は有効であると考えられる。一方で、最適化開始からしばらくの軌道更新周期では、優良でない軌道によって飛行が行われてしまうことが懸念され、目標地点へ到達できなくなってしまう可能性がある。これらの問題点について、シミュレーションによってどの程度の影響があるかを検証するとともに、必要となるコンピュータの性能と軌道更新周期について述べる。

4.2.FPGA を用いた GA によるオンボードリアルタイム軌道

計算

4.2.1.FPGA

FPGA (Field Programmable Gate Array) とは 1 個の LSI の中の配線を設計者が望むようにつなぎ替えることで、目的の論理回路を自由に制作できる回路である[23]. FPGA は、目的の論理回路によって構成されたハードウェア演算により、高速演算および並列演算を実現できることが大きな利点である。これらの特長を生かし、多数の入出力を持つ回路、基板サイズ削減や計算負荷の大きな演算に用いられている。

図 4.2-1 に FPGA 内部構成のイメージを示す。FPGA はハードウェアの最小単位である Logic cell などを接続する配線、スイッチ、入出力ブロック、値を一時的に保持する Block RAM から構成される。

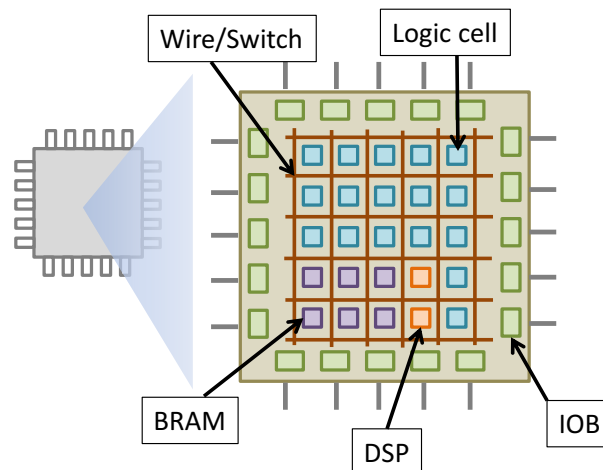


図 4.2-1 Internal Constitution of FPGA

4.2.2.実飛行環境下における軌道生成実証飛行

GA を用いた誘導軌道生成システムについて、シミュレーションによって有効であることは従来研究より示された。故に、実飛行中に実際に軌道生成を行い、優良な最適軌道の生成が行えるかの実証飛行試験を行った。

本アルゴリズムの対象は滑空飛行フェーズの有翼ロケットであるが、ロケットの打ち上げ回数は限られているため、比較的容易に実験が行えるカイトプレーンと呼ばれる低速で遠距離操作が行える小型無人飛行機を用いて飛行試験を行った。図 4.2-2 に飛行試験で使用したカイトプレーンの外観を示す。今回使用したカイトプレーンは株式会社スカイリモートが開発したものである。

飛行自体はこのカイトプレーンを使用するが、誘導システム内部で行う最適化計算は、小型有翼ロケット実験機 WIRES#014 のモデルを使用する。WIRES#014 を図 4.2-3 に示す。また、それぞれの機体諸元を表 4.2-1 と表 4.2-2 に示す。



図 4.2-2 カイトプレーン



図 4.2-3 小型有翼ロケット実験機 WIRES#014

表 4.2-1 カイトプレーン機体諸元

Body Mass	[kg]	18
Length	[m]	2.28
Width	[m]	2.78
Hight	[m]	1.13

表 4.2-2 小型有翼ロケット実験機 WIRES#014 機体諸元

Body Mass	[kg]	45
Wing Area	[m ²]	0.47
Total Length	[m]	1.7

小型有翼ロケット実験機 WIRES#014 は有翼飛翔体として、航法誘導制御システムの飛行実証を行う実験機である。機体は炭素繊維強化プラスチック製モノコック構造で、空力舵面は主翼にエレボン、2 枚の尾翼にラダーを装備している。エンジンは HyperTEK 社の M 型ハイブリッドロケットモータを使用しており、到達高度はおよそ 1000[m] である。回収システムは、2 段式パラシュートシステムを採用しており、減速シュートによって機体速度を殺した後、メインシュートによってゆっくりと降下する。最終的にエアバッグを展開し、無傷で回収することを目的としている。

4.2.3. カイトプレーンに搭載する試験装置

飛行試験で使用したカイトプレーンは、自立飛行制御装置が搭載されている。このシステムは、予め決められた複数のポイントを基準に、GPS センサを用いて次に向かうポイントとの距離および方位を算出し、ジャイロセンサなどにより補正をかけながら機体の舵を取る。全てのポイントを飛行し終わると、再び最初のポイントから飛行を始める。各ポイントは緯度経度のみが決められており、高度に関しては一定の決められた高度を飛ぶように設定されている。

飛行ポイントの設定は、飛行開始前に基地局で行う。基地局にてカイトプレーンに搭載された GPS から位置情報を取得し、そこを原点とする。図 4.2-4 に示すように、各ポイントは原点との方位、距離および相対高度により設定される。基地局では、モニタリングソフトによって現在の飛行状況を監視できる。カイトプレーンに搭載された自律飛行制御装置内の位置情報および制御情報はデータ送信機を解して基地局に送られる。基地局では送られてきたデータをモニタリングソフトで表示を行い、それらのデータをログとして保存する。モニタリングソフトは各ポイント位置、制御の各種設定値を自律飛行制御装置に送り、自律飛行の支援も行う。

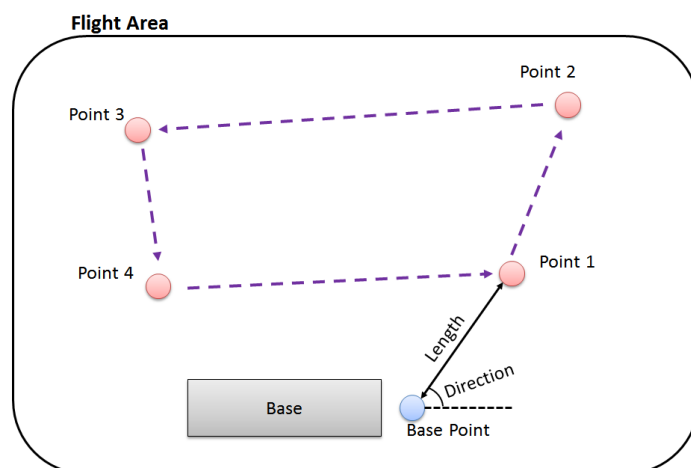


図 4.2-4 カイトプレーン飛行イメージ

カイトプレーンに搭載し軌道生成計算を行うため、各種データの取得および初期条件生成の計算を行う ADS/IMU/GPS 複合航法システムの評価用装置として開発したプロトタイプデバイスの構成図を図 4.2-5、誘導システム内部の構成を図 4.2-6 に示す。プロトタイプデバイスは、複合航法に必要なセンサとセンサから送られてくる情報を処理する各マイコンで構成されている。FPGA（誘導システム）は ADS CPU と RS232C で通信を行い、複合航法システムが取得したデータから軌道の生成を行う。使用する各データは、GPS から送られてくる緯度および経度、IMU から送られてくる Quotation と迎角から経路角および方位角、ADS からは速度と高度である。今回 GPS 高度を使用しないのは、ADS から取得した気圧高度の方が高精度であることが過去の有翼ロケット実験においてわかっているためである。

軌道生成は 1.5 秒ごとに行う。プロトタイプデバイスは表 4.2-3 に示すようなデータを 32-16bit 固定小数点数に加工し、16 進数表記の ASCII データとして誘導システムに送信し、誘導システムは良好軌道生成の可否にかかわらず、生成した軌道をプロトタイプデバイスに表 4.2-4 に示すようなデータで送信する。プロトタイプデバイスは送信されてきたデータを SD カードに保存する。

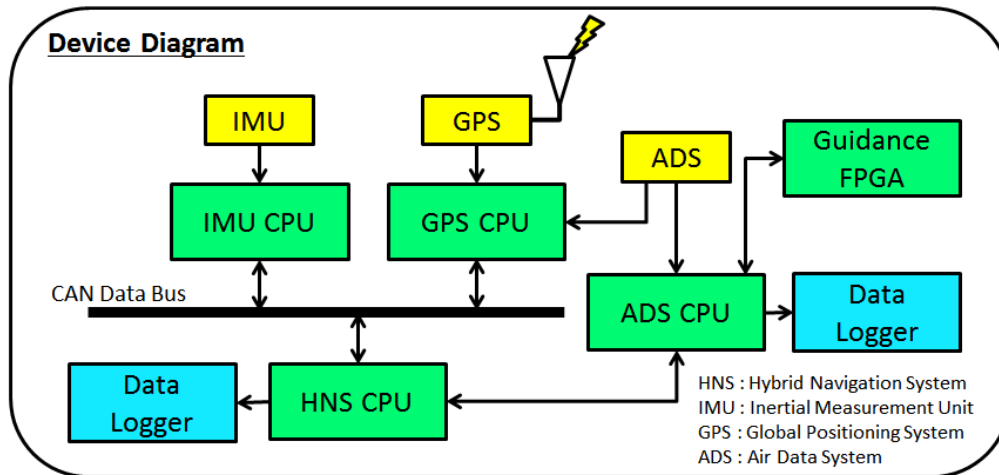


図 4.2-5 プロトタイプデバイスシステム図

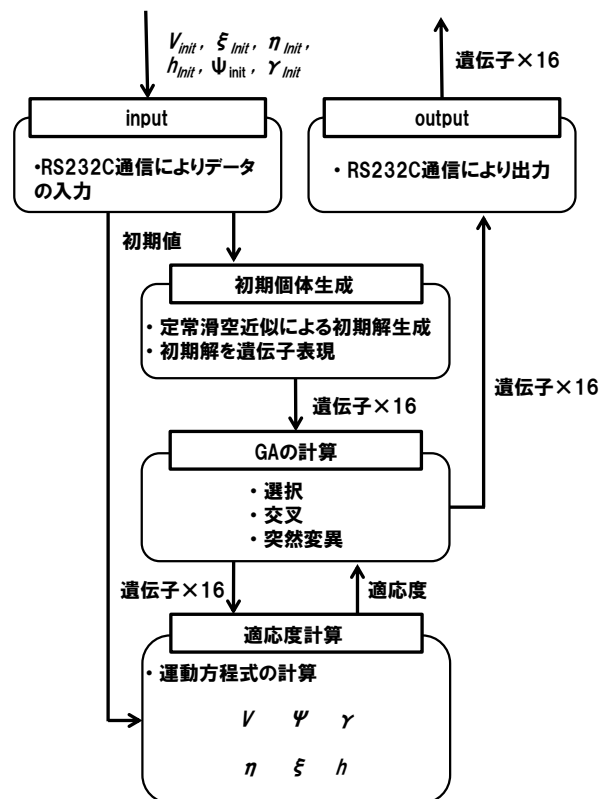


図 4.2-6 誘導システム内部の構成

表 4.2-3 誘導システムへ入力するデータ

	Data	Description
1	I_V	Initial Velocity
2	I_gamma	Initial Flight Path Angle
3	I_psi	Initial Direction Angle
4	I_xi	Initial Down Range
5	I_eta	Initial Cross Range
6	I_alt	Initial Altitude

表 4.2-4 誘導システムから出力されるデータ

	Data	Description		Data	Description
1	gene00	α_{a0}	13	gene12	φ_{b1}
2	gene01	α_{a1}	14	gene13	φ_{b2}
3	gene02	α_{a2}	15	gene14	φ_{b3}
4	gene03	α_{a3}	16	gene15	φ_{w0}
5	gene04	α_{b1}	17	max_fit	Max. Fitness
6	gene05	α_{b2}	18	time	Calculation Time
7	gene06	α_{b3}	19	I_V	Initial Velocity
8	gene07	α_{w0}	20	I_gamma	Initial Flight Path Angle
9	gene08	φ_{a0}	21	I_psi	Initial Direction Angle
10	gene09	φ_{a1}	22	I_xi	Initial Down Range
11	gene10	φ_{a2}	23	I_eta	Initial Cross Range
12	gene11	φ_{a3}	24	I_alt	Initial Altitude

4.2.4. 飛行試験場所

飛行試験場所は熊本県熊本市西区新港にある熊本港フェリー乗り場北側の工業用地内である。試験場所の様子を図 4.2-7 に示す。カイトプレーンでの飛行は表 4.2-5 のように高度が最大 200[m]、飛行速度が最大 20[m/s]である。誘導軌道生成システムでは、有翼ロケットモデルが入力されているため、飛行環境が大きく異なる。そこで取得データから軌道生成可能とするため、有翼ロケットの飛行環境へ沿うように気圧高度に 1500[m]、速度に 50[m/s]のオフセットを設けて計算初期条件とする。

軌道生成におけるダウンレンジ、クロスレンジの原点を北緯 32° 45' 49.4", 東

経 130° 35' 6.2"として、この地点の上空 400[m]を目標地点とする．ここでダウンレンジ正方向を北，クロスレンジ正方向を東と設定する．軌道生成計算は枠に囲まれた領域をカイトプレーンで飛行し軌道の生成を行う．

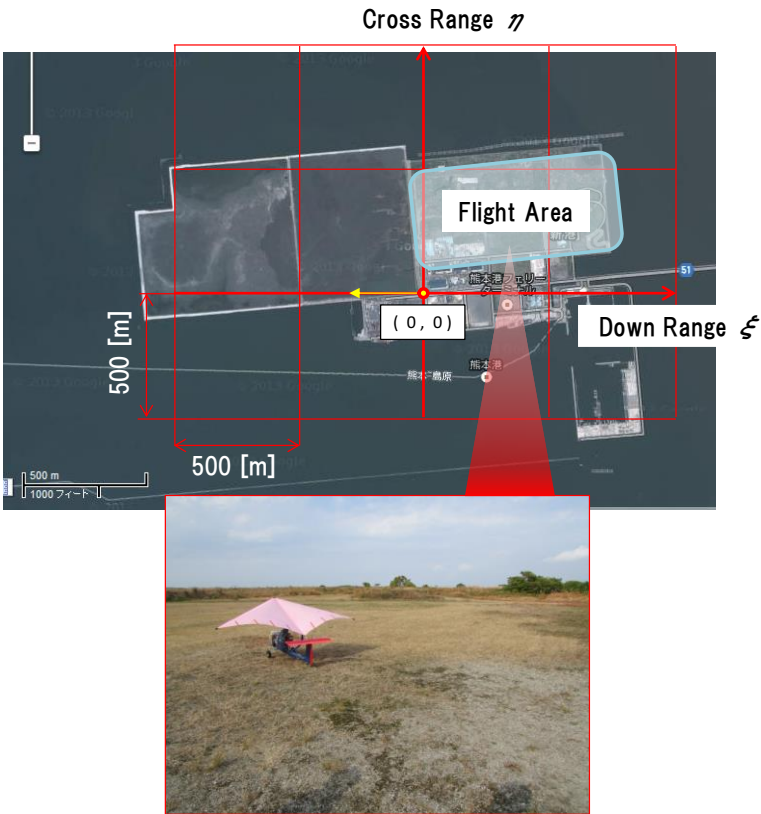


図 4.2-7 飛行試験実施場所外観

表 4.2-5 カイトプレーン飛行環境とシミュレーション使用値

		Kite Plane	Offset	Expected range
Velocity	[m/s]	0～20	+50	50～70
Altitude	[m]	0～150	+1500	1500～1650

4.2.5.GA 計算条件

本飛行試験に限り，GA 計算条件を表 4.2-6 のようにした．

表 4.2-6 カイトプレーン飛行試験の GA 計算条件

Populations	24
Generations	250
Change Model	Elitist Recombination
Mutation Rate	5%
Crossover	Blend Crossover(BLX-0.5)
Equation of Motion	Heun's Method
Step Width of Calculation	1 [s]
Control Inputs	Angle of Attack, Bank Angle (Fourier Series of Third Order)

また、初期個体生成時には、予め優良な個体を生成しておくことにより、最適化に必要な計算時間の短縮を図った。その概要を図 4.2-8 と図 4.2-9 に示す。ここで、 α は迎角、 δ はスピードブレーキ角、 ϕ はバンク角を表し、添字 *max* はそれぞれの舵角の最大値を表す。

初期条件と目標条件の関係と定常滑空近似より、円と直線を用いて目標地点へ到達できる飛行軌道を幾何学的に推定する。このとき、各円の旋回半径は、バンク角最大時の旋回半径、つまり最小旋回半径とする。これにより、バンクをする時間と水平飛行をする時間の推定が行え、コマンド履歴を推定できる。ここから、フーリエ変換を行い、その係数に一定の乱数をかけた値を初期個体としている。

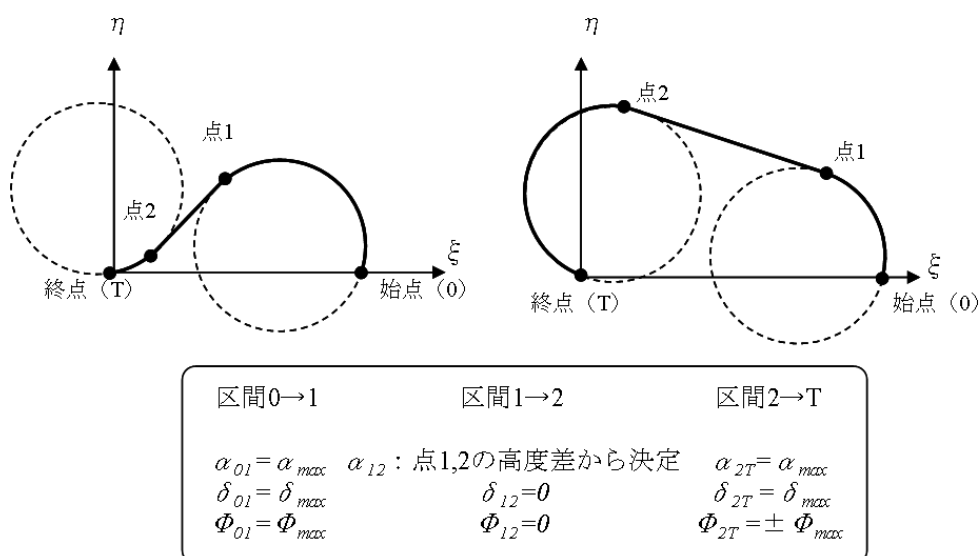


図 4.2-8 定常滑空近似概要

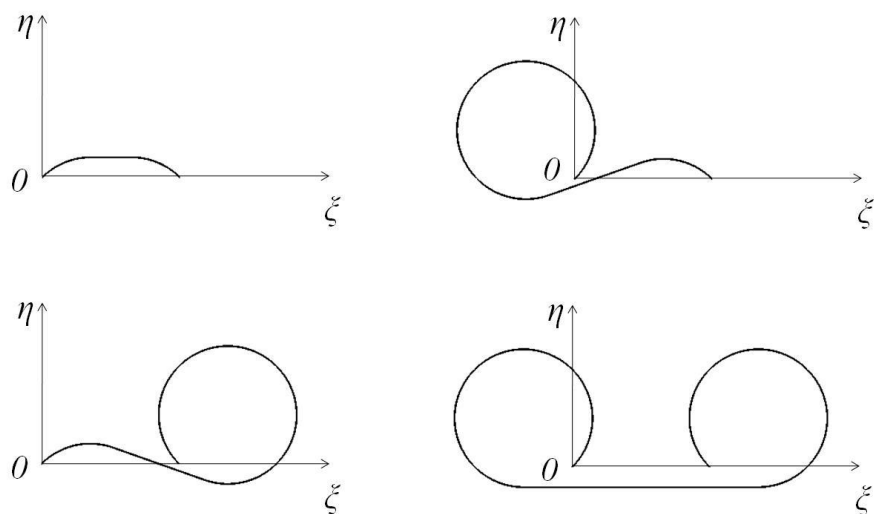


図 4.2-9 定常滑空近似から得られる軌道例

誘導システムを実装した FPGA の外観と諸元を図 4.2-10 と表 4.2-7 に示す.



図 4.2-10 FPGA 外観

表 4.2-7 FPGA 諸元

Model Number	Xilinx Spartan-6 XC6SLX150-2FGG484C
Slice	23,038
Logic cell	147,443
DSP	180
Onboard Oscillator	30 / 50 [MHz] (Using 50[MHz])
I/O Port	100 Ports
Board Size	54[mm] × 86[mm]

4.2.6. 飛行試験結果

飛行試験は 2013 年 12 月 1 日に全 5 回行った。その飛行試験結果の一部を図 4.2-11 と図 4.2-12、表 4.2-8 に示す。これらの結果より、初期高度を 1500[m]とした場合、ほとんどの初期条件において、平均 1[s]で良好な軌道をオンボードで生成できることが実証できた。一部で特異解が生成されているが、初期位置の高度が目標高度を下回っていたり、初期経路角が 60 度を超える滑空飛行とは言いがたい条件となっていたりする場合であった。

表 4.2-8 カイトプレーンによる飛行実証結果

Test	Number of Generated Trajectories	Average Maximum Fitness	Average Calculation Time [s]	Number of Failed Trajectories
1	251	16.1	0.872	12
2	161	13.0	0.964	30
3	93	21.2	0.949	1
4	180	15.7	0.917	4
5	433	15.7	0.908	24

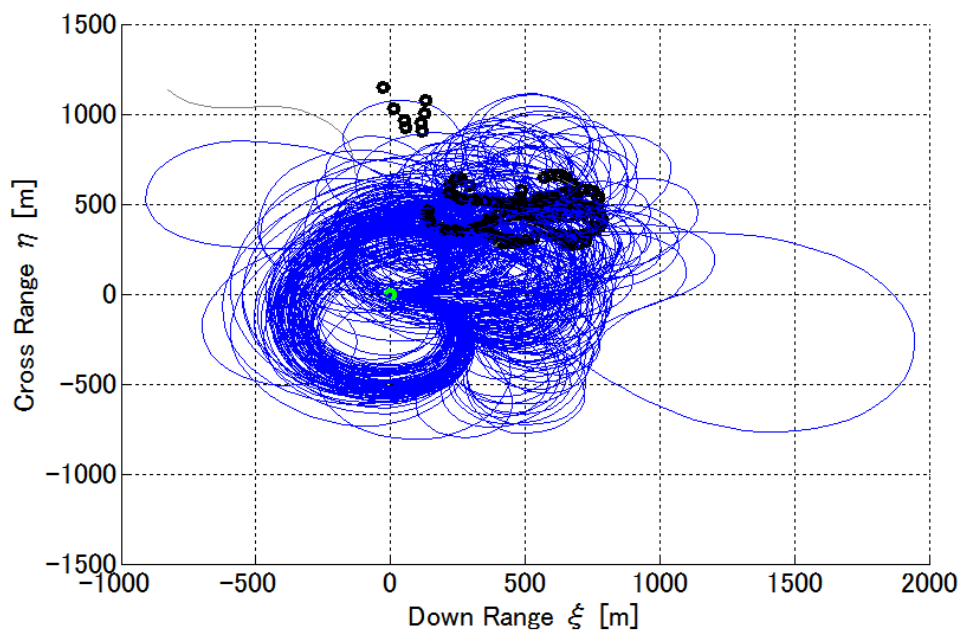


図 4.2-11 カイトプレーン飛行試験時に生成された誘導軌道（平面図）

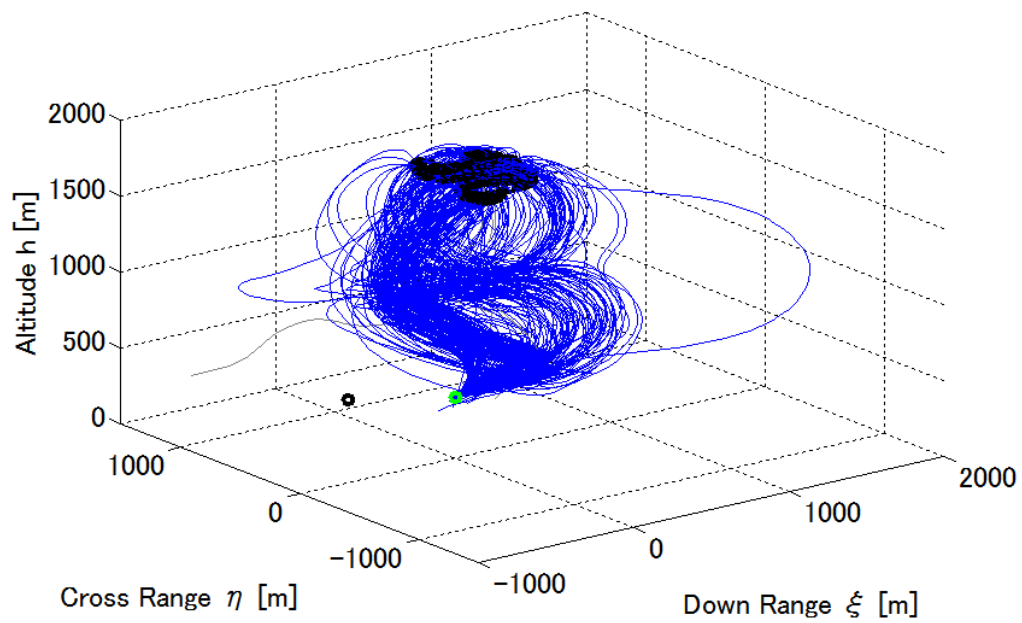


図 4.2-12 カイトプレーン飛行試験時に生成された誘導軌道（鳥瞰図）

4.3. DynDGA を用いた誘導軌道再生成シミュレーション

4.3.1. 誘導軌道再生成シミュレーション

これまでの誘導軌道生成シミュレーションによって、飛行禁止区域を有する飛行領域においても目標地点へ到達できる複数の誘導軌道生成に成功した。また、オンボードリアルタイム軌道生成においても、飛行中のデータから目標地点へ到達する軌道が得られることが解った。しかし、これらの実証では軌道の生成のみを単発で行うことしかしておらず、実際にその軌道に沿って飛行した場合のこと、さらにはその軌道に沿って飛行している最中に再度誘導軌道を生成した場合における挙動等、本誘導システムの検証が行えていない。

そこで、サブオービタル飛行を行う有翼ロケットの滑空帰還フェーズを対象に、DynDGA による誘導軌道生成システムを用いた飛行シミュレーションを行った。このシミュレーションでは、誘導システムの評価だけでなく、制御システムの評価を行うため、アクチュエータの応答モデルの設計や制御ゲインの調整などの作業が行われたが、本論文では誘導システムについてのみ評価を行う。

4.3.2. シミュレーションの構成

図 4.3-1 にシミュレーション処理の概要を示す。本シミュレーションでは、打上から回収フェーズまでのシミュレーションとなっている。図 4.3-2 に示すような飛行フェーズになっており、DynDGA を用いた誘導軌道生成システムは、滑空帰還フェーズのみを対象としている。

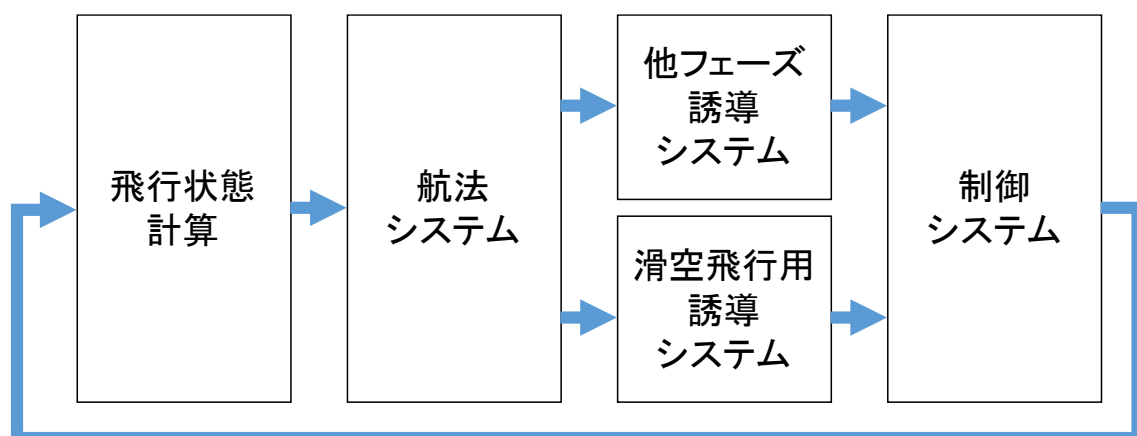


図 4.3-1 誘導軌道再生成シミュレーション概要

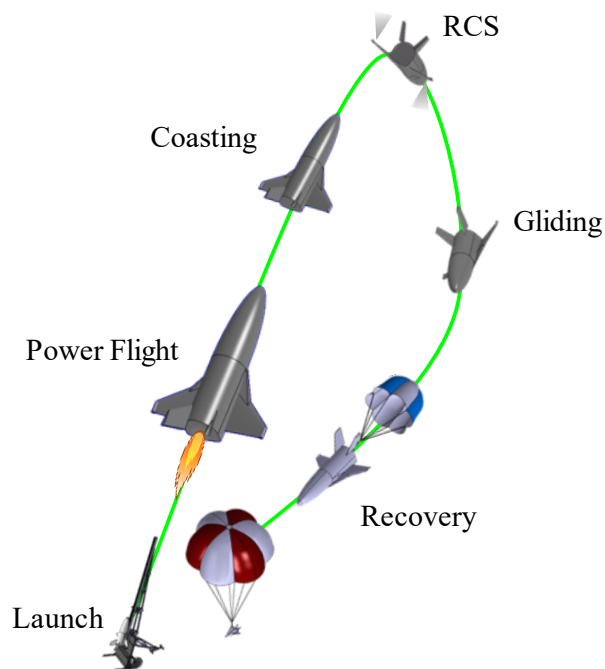


図 4.3-2 誘導軌道再生シミュレーションのフェーズ

各フェーズにおける誘導コマンドの種類とフェーズ以降の条件を図 4.3-3 に示す。基本的に滑空帰還フェーズ以外は各コマンドを時系列データとして与えている。滑空帰還フェーズは、DynDGA を用いた軌道生成システムを用いて、現在の飛行環境から誘導軌道の生成を行い、得られた解から誘導コマンドを出力している。

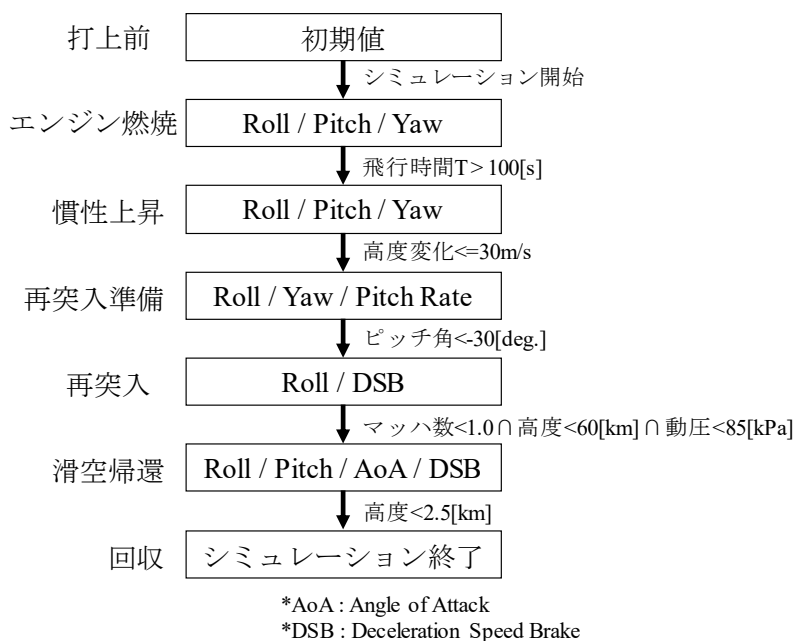


図 4.3-3 各フェーズにおける誘導コマンド

制御則に関しては，上昇から再突入までを PD 制御によるフィードバック制御を行い，滑空帰還フェーズのみダイナミックインバージョンを用いた制御を行っているが，詳細については省略する．ダイナミクス計算は，古典的ルンゲクッタによる 6 自由度の運動方程式を積分している．

図 4.3-4 滑空帰還フェーズ中の誘導システム処理概要に滑空帰還フェーズ中の誘導システムの処理概要を示す．DynDGA を用いた誘導軌道生成システムは，軌道の更新周期に設定した 5[s]毎に，5 秒後のロケットの位置を推定し，その位置を初期条件として最適化，新しい誘導軌道を生成する．滑空飛行中は生成された誘導軌道の姿勢コマンド（迎角，バンク角，スピードブレーキ角）に従って飛行を行う．DynDGA の条件は表 2.3-1 と同様である．

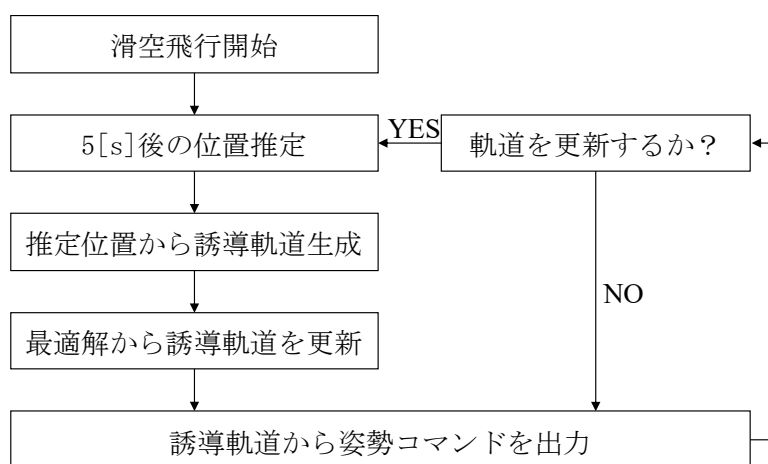


図 4.3-4 滑空帰還フェーズ中の誘導システム処理概要

4.3.3.機体モデル

本シミュレーションで用いた機体モデルを図 4.3-5 と表 4.3-1 に示す．これまでの機体モデルとは異なり，スピードブレーキとボディフラップ，RCS ガスジェットすらスターを搭載しており，再突入フェーズ前後は，これらによって機体の姿勢やスピードの制御を行っている．空力係数は，HIMES の空力パッケージを利用している（図 4.3-6）．各アクチュエータの動作範囲は表 4.3-2 の通りである．

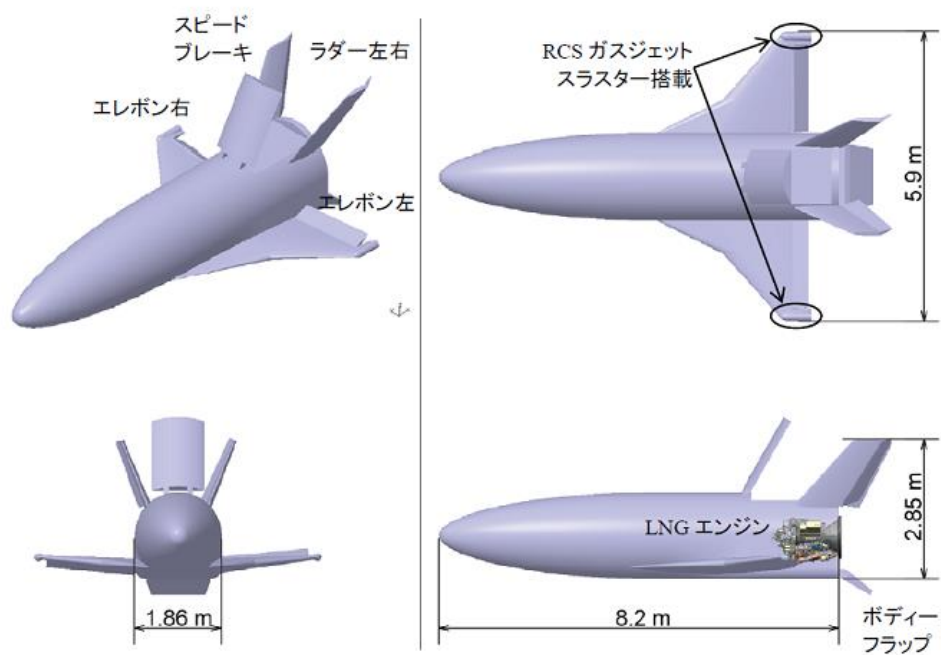


図 4.3-5 誘導軌道再生成シミュレーションの機体モデル

表 4.3-1 誘導軌道再生成シミュレーションの機体諸元

m	Body Mass	1734	[kg]
S_w	Wing Area	11.26	[m ²]
S_b	Speed Brake Area	0.12	[m ²]

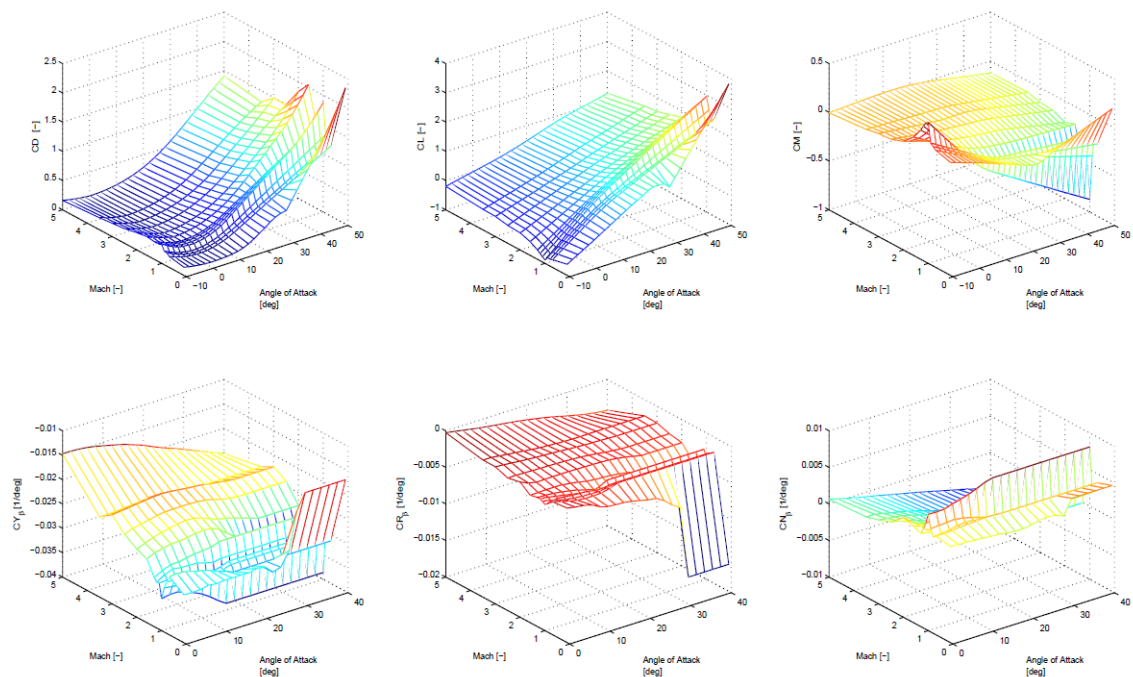


図 4.3-6 HIMES 空力パッケージ Ver. 1.4 基本空力係数

表 4.3-2 各舵角範囲

エレボン左舵角	±30[deg.]
エレボン右舵角	±30[deg.]
ラダー左舵角	±30[deg.]
ラダー右舵角	±30[deg.]
スピードブレーキ舵角	0～+90[deg.]
ボディフリップ舵角	±30[deg.]
RCS 指令	0：バルブ閉, 1：バルブ開
エンジン指令	0：燃焼停止, 1：燃焼中・点火

4.3.4. 飛行領域

飛行領域を図 4.3-7 に示す．打ち上げ場所は Spaceport America 付近 (北緯 32.94033[度]東経-106.9066[度])を想定し，回収目標地点は打ち上げ場所から東へ約 10[km]離れた地点 (北緯 32.94028[度]東経-106.7996[度])とする．飛行禁止区域等の設定は無く，無風とする．また，着陸模擬評価を目的として，回収目標点上空 2.1[km]に仮想滑走路を想定する．

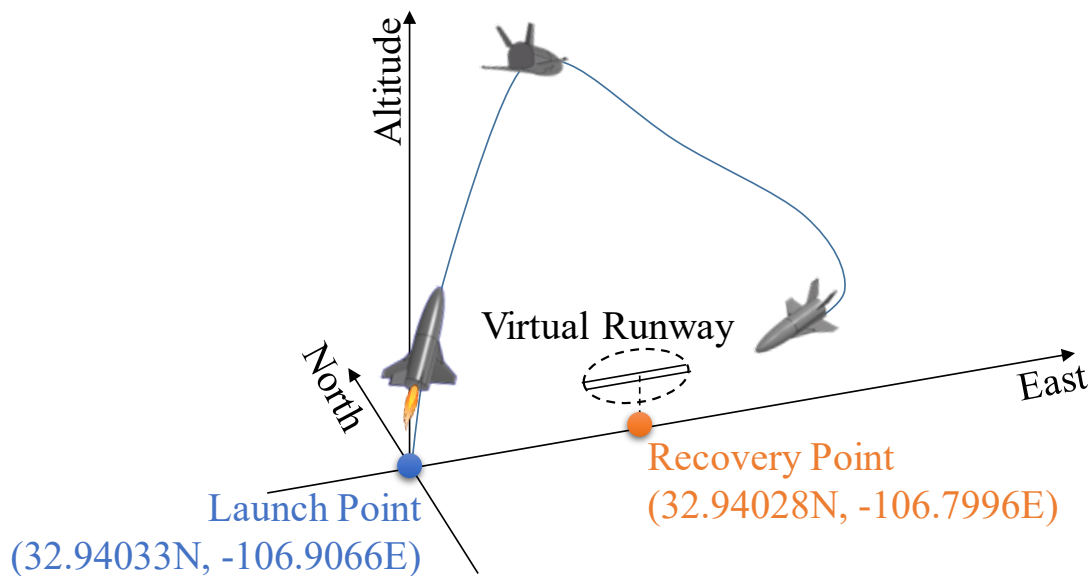


図 4.3-7 誘導軌道再生成シミュレーションの飛行領域

4.3.5. シミュレーション結果

図 4.3-8～図 4.3-13 にシミュレーション結果を示す．突入フェーズ後，有翼ロケットは目標地点へ滑空飛行を行い，目標高度でほぼ目標地点へ到達した．姿勢角履歴より，目標地点付近で振動してしまう結果となったがほぼシミュレーションは成功したと言える．一方で，本飛行シミュレーションを行うに当たり，1回の誘導軌道生成に約 300 秒と膨大な時間がかかってしまうことが解った (Windows 8.1, Core i-7 3.2GHz, 並列無)．この計算時間は，第 2 章および第 3 章で行った計算時間と比較しても数倍と膨大な時間である．その理由として，各個体の適応度を計算するに当たり，運動方程式を積分して軌道の計算を行っているが，この軌道の計算は積分ステップ数が多いほど時間がかかってしまうことが考えられる．この積分ステップ数は，飛行時間に比例し，飛行時間は初期位置と目標位置との高度差と初期位置における方位角に依存している．故に，飛行高度が高く，飛行距離が長いほど，最適化には時間を要してしまうことがわかった．

制御入力についても課題が見られた．軌道を再生成するに当たり，前回の最適化の結果を適応度の高い解から半分，引き継いだ状態 (残りの個体はランダム生成) で最適化を行っているが，制御入力の値に関して，誘導軌道更新周期における現在のロケットの姿勢角と，再生成された誘導軌道の姿勢角コマンドとの間には特に制限を設けておらず，急激にコマンドが変化してしまうことが解った．これについても，対策をとる必要があると考える．

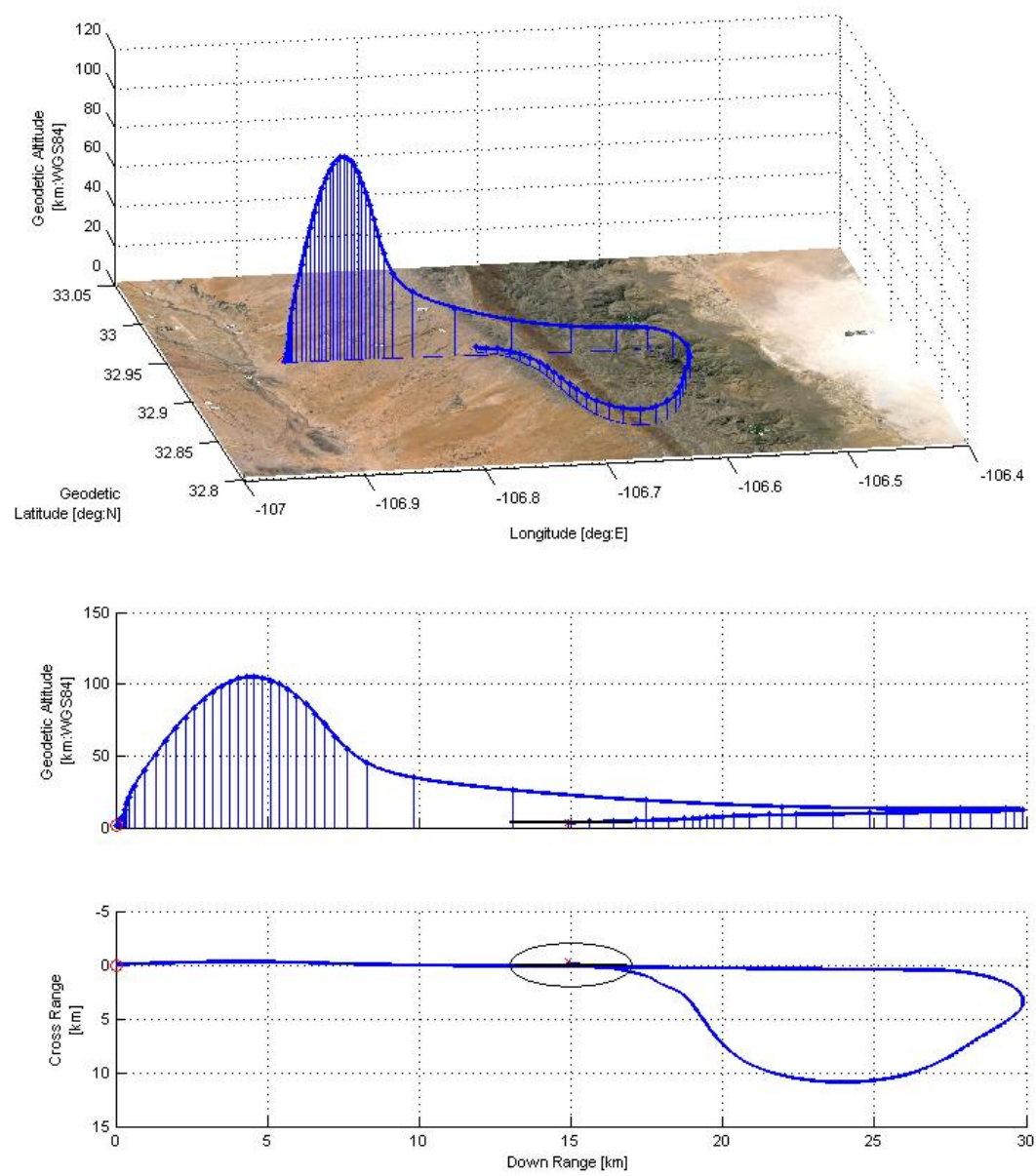


図 4.3-8 誘導軌道再生成シミュレーション結果 A

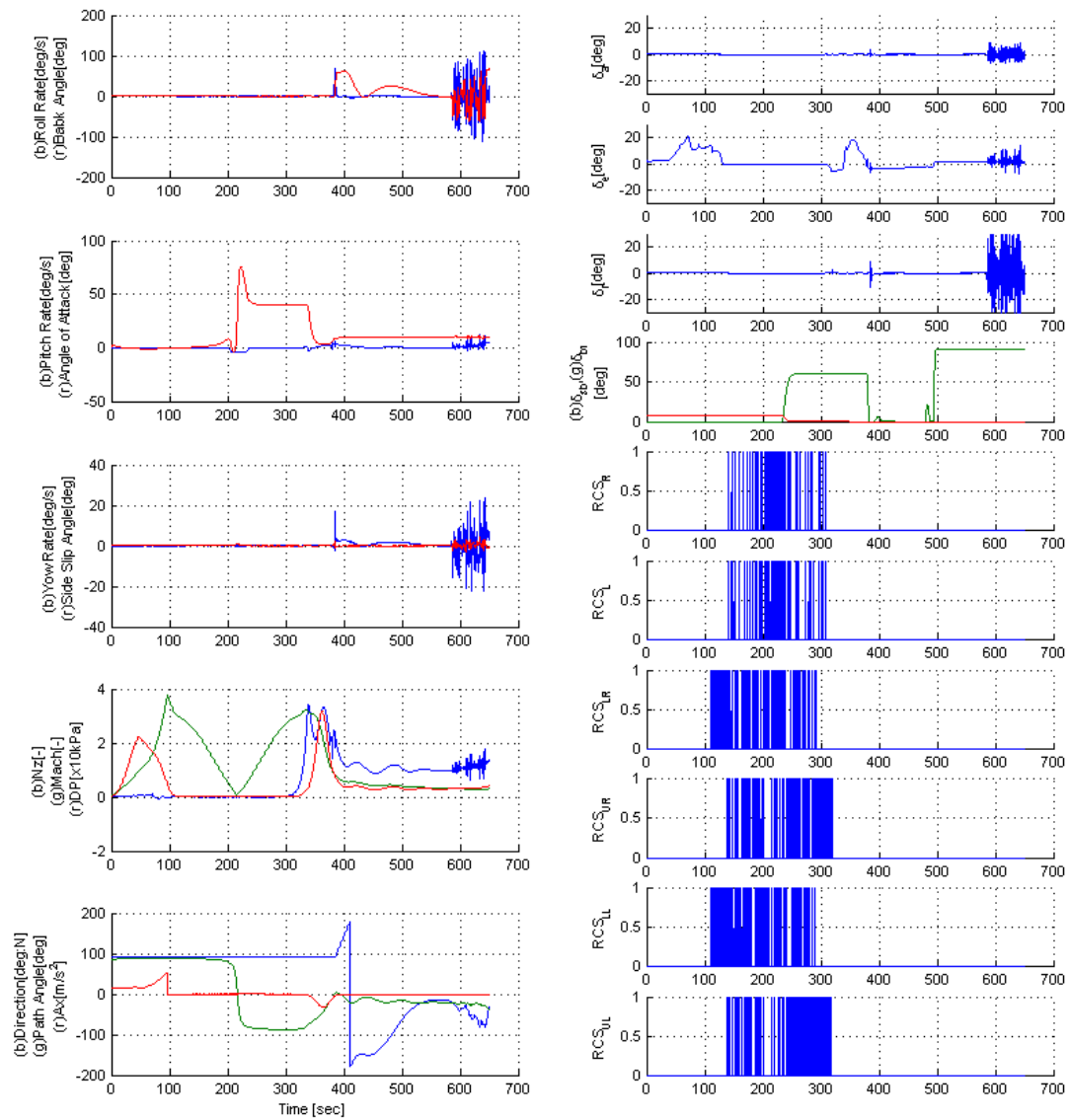


図 4.3-9 誘導軌道再生シミュレーション結果 B

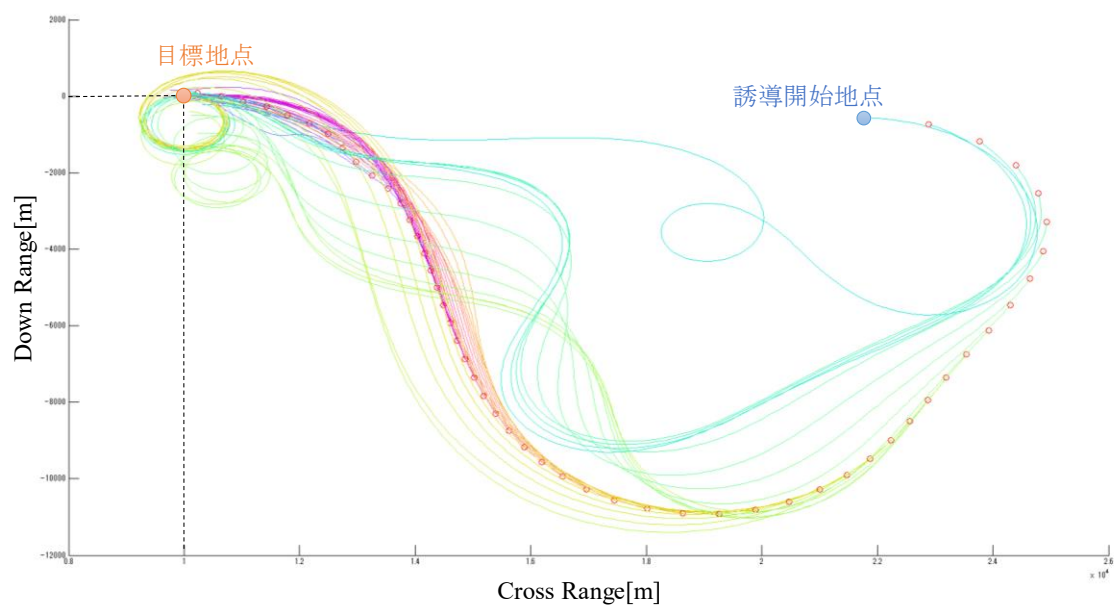


図 4.3-10 誘導軌道再生成の様子

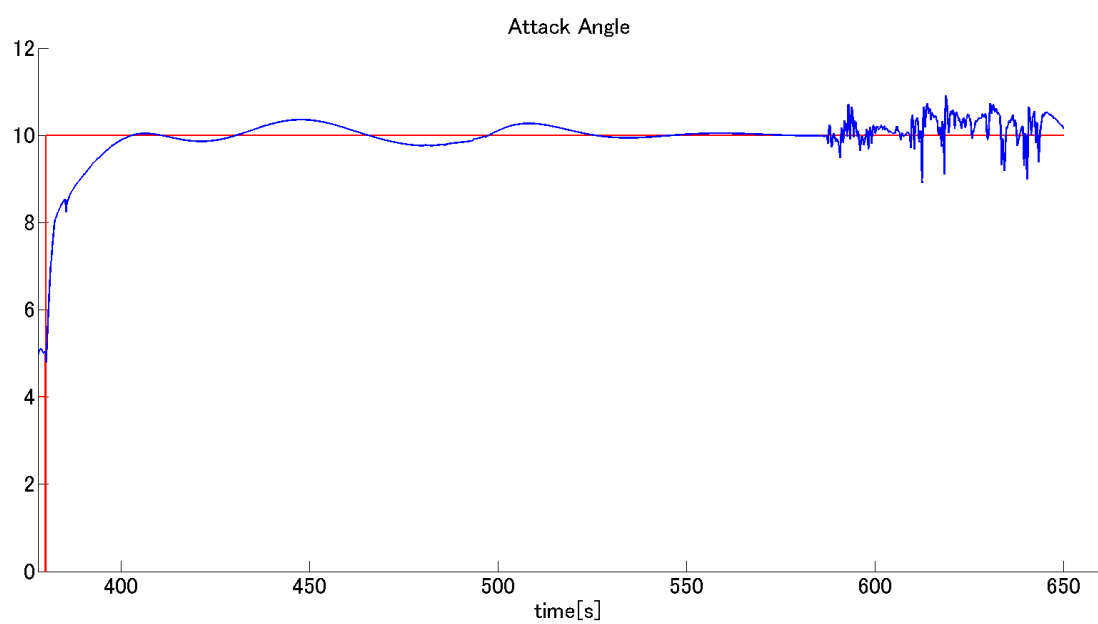


図 4.3-11 迎角コマンド(赤)と飛行結果(青)

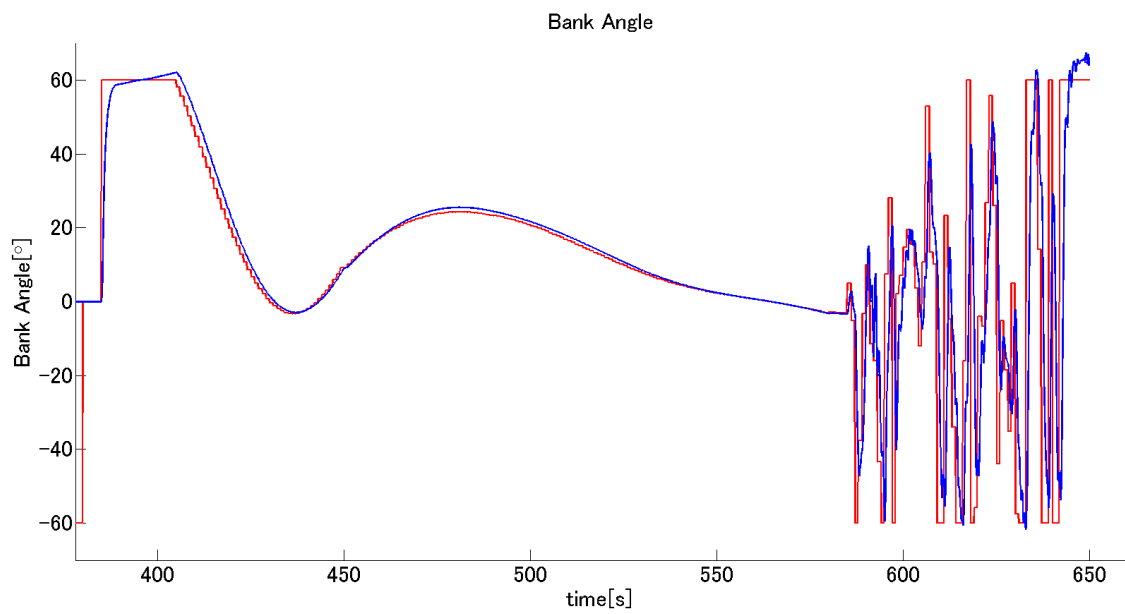


図 4.3-12 バンク角コマンド(赤)と飛行結果(青)

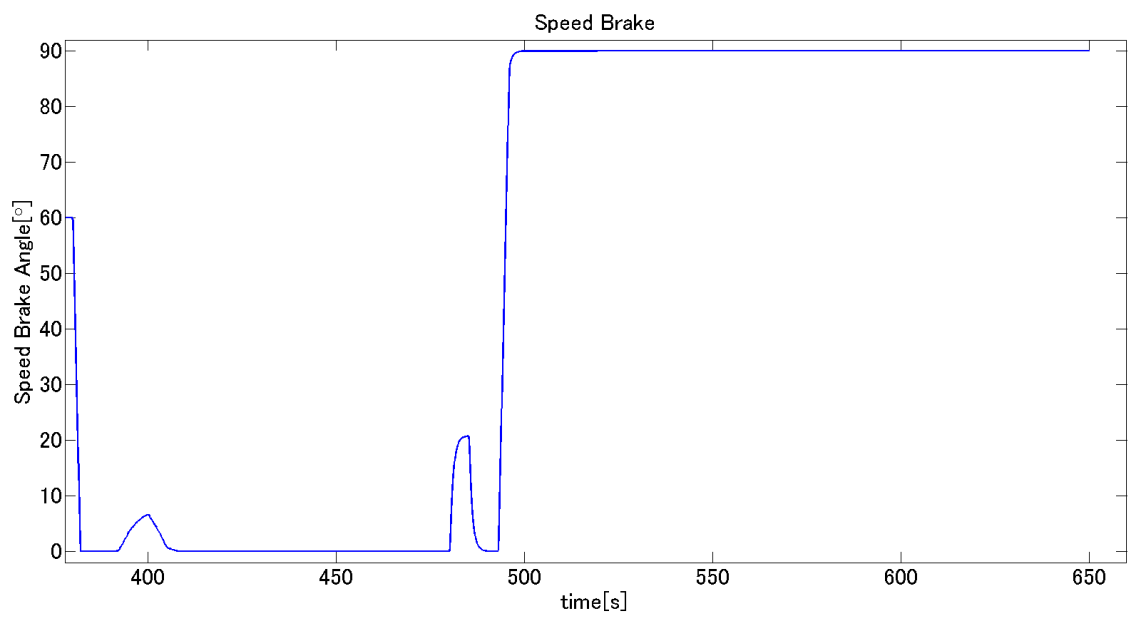


図 4.3-13 スピードブレーキコマンド(赤)と飛行結果(青)

4.4. DynDGA を用いたリアルタイム誘導軌道生成システムの検討

4.4.1. 実飛行を考慮したリアルタイム化の検討

4.3.5 にて述べたように、DynDGA を用いた誘導軌道生成には、飛行距離に応じた計算時間が必要である。しかし、実際の飛行では、軌道を再生成することはもちろんのこと、アボート飛行なども考慮した場合、飛行距離を事前に想定するのは難しい。高度 5[km]まで飛行する WIRES#015 モデルにおいて、リアルタイムで誘導軌道をオンボード生成するには、4.2 で述べた FPGA の計算条件を参考に、以下のように計算した。表 4.4-1 には計算条件の比較を示す。

表 4.4-1 計算条件比較

	WIRES#014 FPGA 実装モデル	WIRES#015 モデル
最適化手法	GA	DynDGA
最大高度差	600[m]	5000[m]
最大距離	1000[m]	10000[m]
個体数	40	200
世代数	200	800

WIRES#014FPGA 実装モデルは、計算時間が最大 1[s]であるので、

$$\text{計算時間} = 1[\text{s}] \times \frac{200}{40} [\text{倍個体数}] \times \frac{200}{400} [\text{倍世代数}] \times \frac{5000}{600} [\text{倍高度差}] = 167[\text{s}]$$

と推定される。さらに、最適化手法が GA から DynDGA に変更しているため、クラスタリングの計算時間がさらに増加する。その計算時間を暫定的に 5[s]とした場合、合計計算時間は 172[s]になる。

WIRES#014FPGA 実装モデルで使用していた FPGA を用いてリアルタイム化を図る場合、8 並列で 43[s], 69 並列で 5[s]となると推定される。表 4.4-2 の様な性能制約のため、FPGA ボード 1 枚につき 5 並列化が可能であることから、5[s]以内に計算が終了するためには FPGA ボードが 14 枚必要であることがわかった。

表 4.4-2 FPGA 並列化における機能使用率

	FPGA	Single	Double	Quintet
CPU		Spartan-6 XC6SLX150		
Clock		30/50[MHz]		
Slice	184314	27640	38301	70284
LUTs	92152	63180	69003	86472
Block RAM	268	6	6	6
DSPs	180	168	168	168

WIRES#015 で使用するには、FPGA が 14 枚あれば、どのような条件下であっても最適化が 5[s]未満で終了すると考えられる。一方で、高価な FPGA が複数枚必要であること、さらに WIRES#015 の性能が向上し、飛行高度が高くなった場合には、さらに FPGA が追加で必要となる可能性がある。

このように、現状のシステムでは十分に余裕を持たせた計算機能力を保有する方法で計算不可の高い DynDGA の最適化を賄っているが、非常に非効率的である。そこで、図 4.4-1 に示すような、飛行中に常に最適化を行うことを前提とし、最適化終了のタイミングを世代数では無く実計算時間で行うことを提案する。

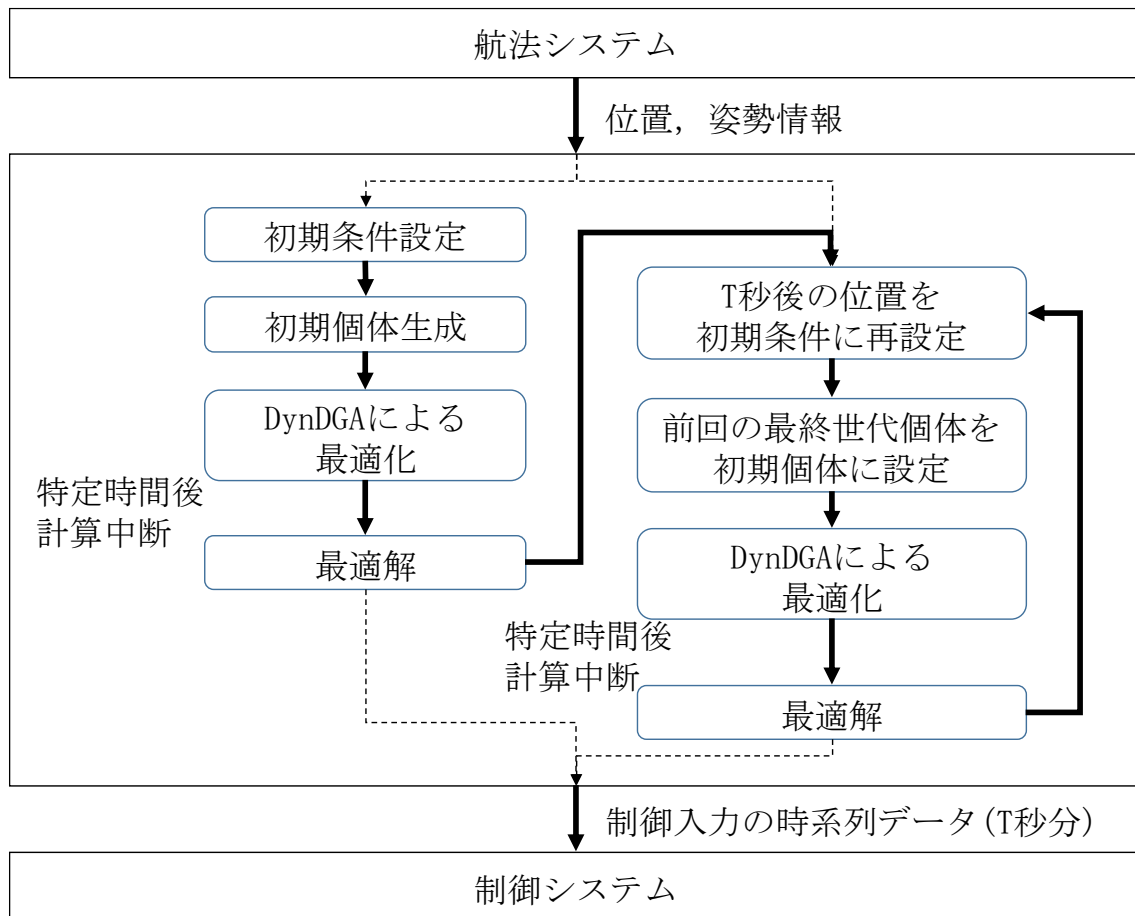


図 4.4-1 リアルタイム性を考慮した誘導軌道生成システム概要

このシステムの鍵は、いかに少ない世代数である程度最適性を持つ解軌道を生成できるかにある。実計算時間で最適化を終了する場合、1世代あたりに必要とする計算時間によって、制御システムへ入力する初回の誘導軌道が、ほとんど最適化が行われる前の解となってしまう可能性があるからである。一方で、最適化は飛行中にも継続して行うため、1世代あたりに必要とする計算時間が長くなる、高度が高い条件から滑空飛行を始める場合、目標値点までの飛行時間も必然的に長いため、滑空飛行の途中で最適化が進み、目標位置へ到達可能な最適軌道が生成されることが考えられる。

以下に、本システムを用いた場合の要点を纏める。

<初期位置と目標位置の高度差が小さい場合>

1世代あたりに必要とする計算時間は短い。故に計算できる世代数が多いため、誘導軌道更新周期毎の適応度は高いと推定される。1回目の誘導軌道生成から目標地点付近へ到達できる軌道生成が期待できる。

＜初期位置と目標位置の高度差が大きい場合＞

1 世代あたりに必要とする計算時間は長い．故に計算できる世代数が少ないため，誘導軌道更新周期毎の適応度は，初期は低いと推定される．1 回目の誘導軌道生成では，目標地点方向へ向かう，といった最適性の低い軌道を制御へ入力することとなるが，トータルの飛行時間が長いため，飛行中に最適化が進み，最終的に目標地点へ到達できる軌道の生成が期待できる．

本誘導システムは，第 3 章で行った階層的手法を用いた適応度計算手法とも親和性が高い．初期位置と目標位置との高度差が大きい場合，解探索が進まず，最適性が低い状態で解を制御へ入力することになってしまう．そこで，解探索において，評価指標の優先度を設定し，階層的手法を用いた適応度計算手法による最適化を行った場合，先に満たしておいてほしい評価指標を優先的に満たしてくれるため，全体の適応度が低い状態であっても，優先度の高い評価指標に限ってはその評価値は高いことが期待できるからである．

4.4.2. リアルタイム性を考慮した誘導軌道再生成シミュレーションの条件

図 4.4-2 と表 4.4-3 にシミュレーションで使った機体モデルを示す．今回のシミュレーションでは WIRES#015 モデルを使用している．

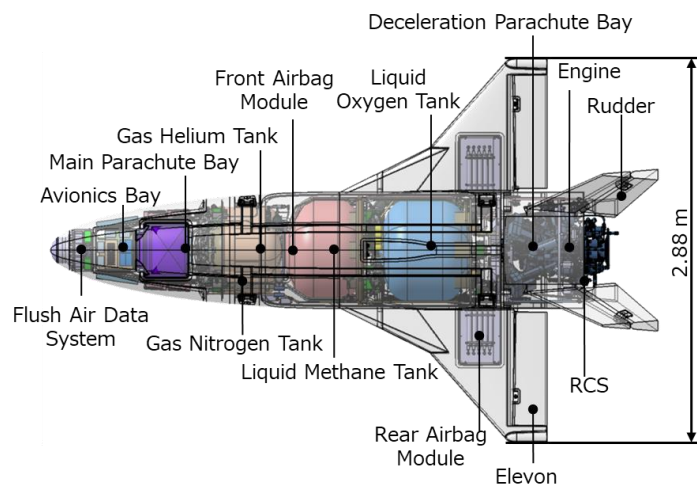


図 4.4-2 WIRES#015 モデル

表 4.4-3 WIRES#015 機体諸元

打上質量	1000[kg]
全長	4.6[m]
最高高度	6[km]
代表翼面積	2.3[m ²]

WIRES#015 はサブオービタル飛行に必要な基礎技術の飛行実証を目的として開発され、20[kN]級の LNG 液体燃料推進系を備えている。

WIRES#015 は空力による機体姿勢制御のため、左右の主翼にエレボンを、2 枚の安定翼にラダーを備える。また、エンジンにジンバル機構を備え、動力上昇中はジンバル機構とエレボン、ラダーの空力舵面のハイブリッドで姿勢制御を行う。RCS も機体エンドフランジに 4 機備え、軌道頂点付近でピッチ角とロール角の制御を行う予定である。なお、本誘導システムの対象である滑空飛行中は、左右のエレボンとラダーのみで姿勢制御を行うものとする。

飛行領域は図 4.4-3 のように定義する。

飛行禁止区域が設定されており、有翼ロケットはこの領域を回避しながら目標地点へ飛行する。評価項目の優先度は表 4.4-4 の様に設定されている。

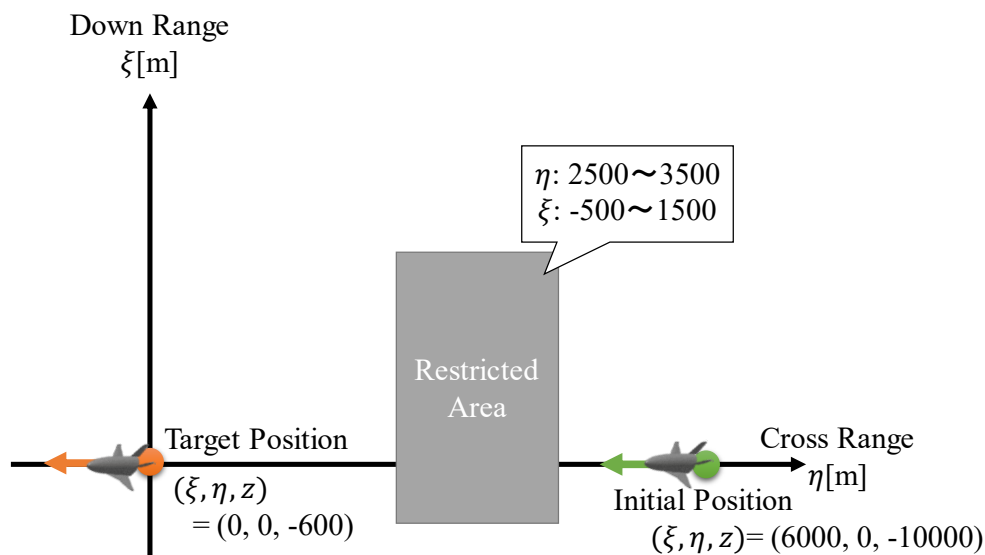


図 4.4-3 誘導軌道再生シミュレーション飛行環境

表 4.4-4 誘導軌道再生成シミュレーションの評価項目優先度設定

優先度	項目
1	飛行禁止区域飛行距離
2	到達地点距離誤差
3	到達地点方位角誤差
4	飛行距離

4.4.3. リアルタイム性を考慮した誘導軌道再生成シミュレーション

結果

図 4.4-4 から図 4.4-8 にシミュレーション結果を示す.

有翼ロケットは目標地点へ到達することが出来ている.

懸念通り, 誘導軌道は初回では目標地点へ到達できていないものの, 飛行途中の最適化によって, 最終的に目標地点へ到達する誘導軌道が得られている.

また, 飛行の途中で誘導軌道の特徴が, 飛行禁止区域のダウンレンジプラス側を飛行しようとする軌道と大きく円弧を描いて飛行しようとする軌道の 2 種類に変化した. これは, 初回の少ない世代数では, 島毎の軌道の適応度が低く, 到達地点へ到達できる軌道が一つも無いため, その世代で最適な軌道を選んだ結果だと考えられる. 一方で, 少ない世代でも, 評価項目の優先度が高い飛行禁止区域を飛行する距離に関して, 全て回避できていると考えられる.

制御入力に関して, 懸念通り, 飛行軌道を再生成するたびに, 大きく値が変化し結果的に不連続となってしまう. これは, 軌道を再生成する際に, 前回の最適化結果から, 次の地点における制御入力の値を考慮に含めずに最適化をしているからであると考えられる.

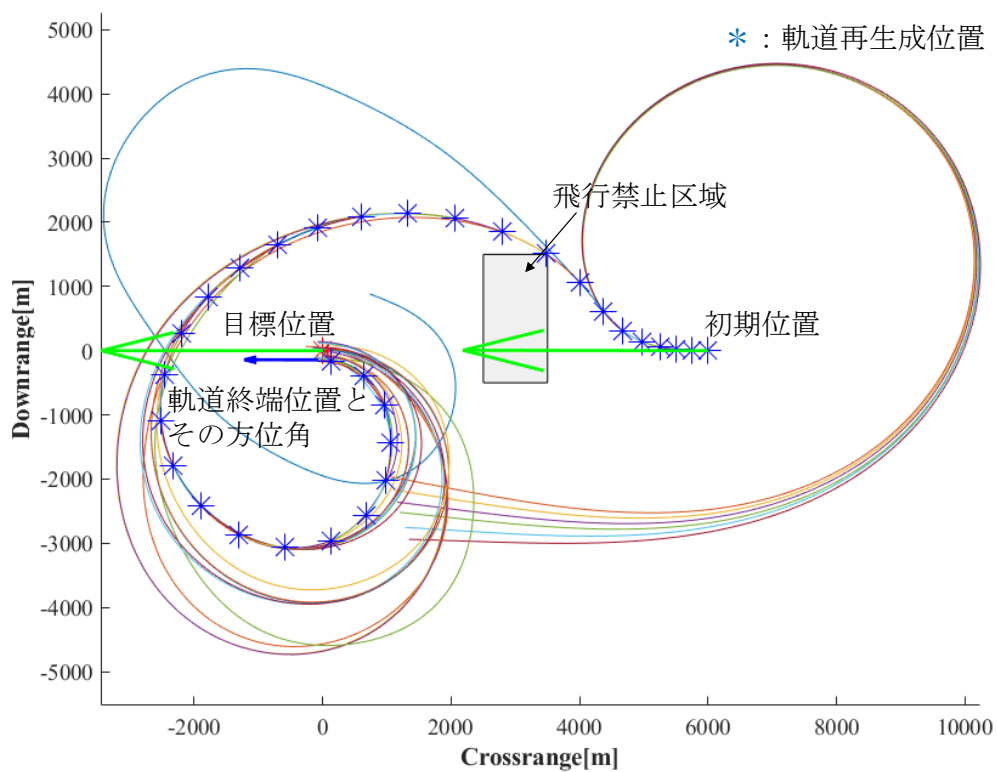


図 4.4-4 再生成シミュレーション結果(飛行軌道)

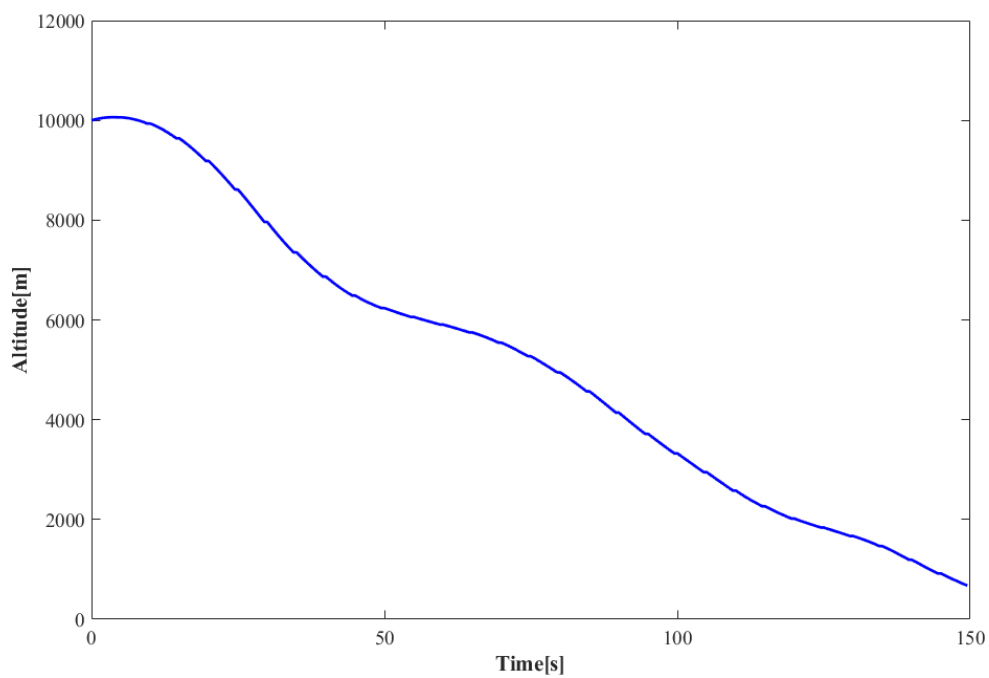


図 4.4-5 再生成シミュレーション結果(高度プロファイル)

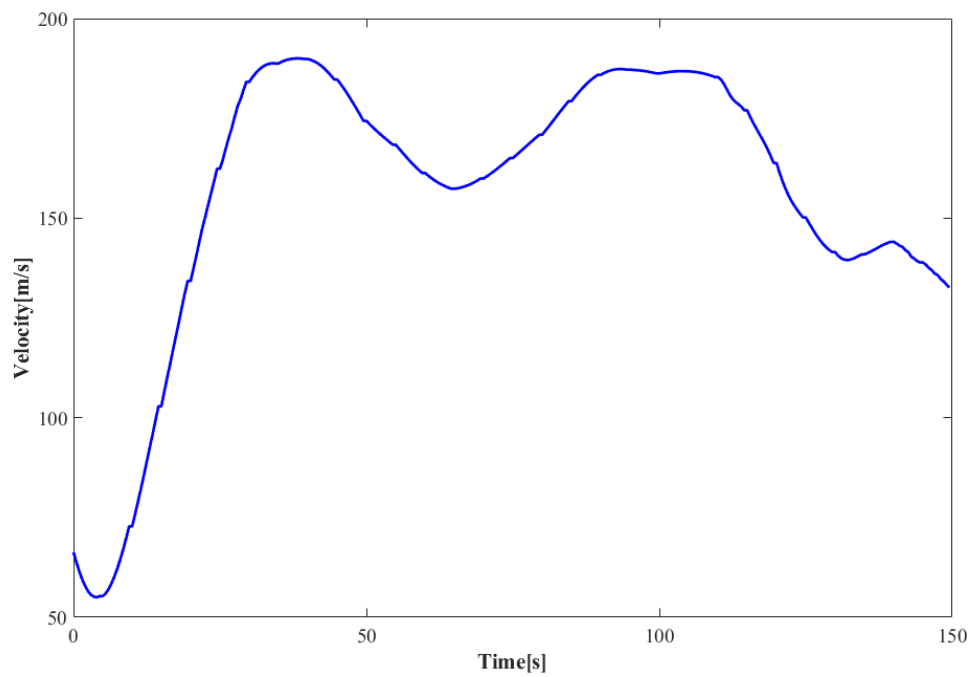


図 4.4-6 再生成シミュレーション結果(速度プロファイル)

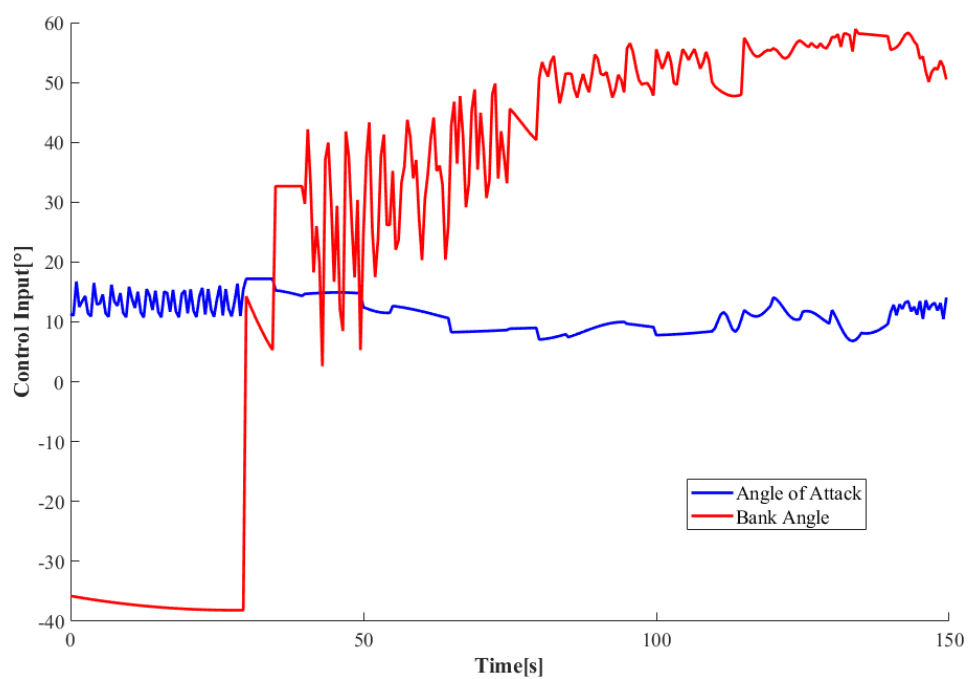


図 4.4-7 再生成シミュレーション結果(制御入力プロファイル)

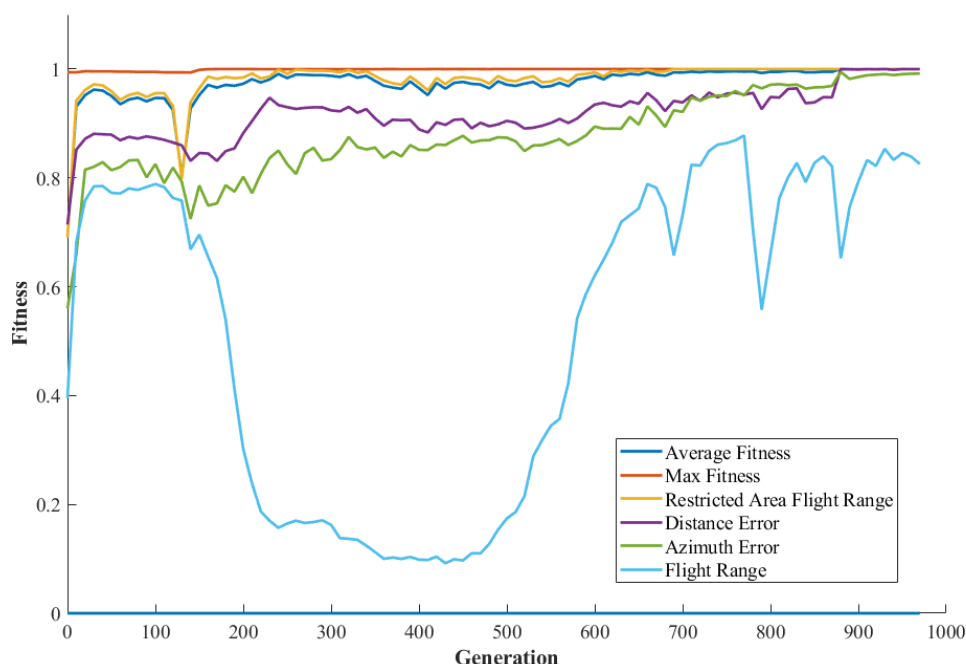


図 4.4-8 再生成シミュレーション結果(メンバーシップ関数値)

4.5. おわりに

実用化に向けて、リアルタイム性を考慮した誘導軌道生成システムは、世代数で最適化計算を終了するのではなく、実計算時間で最適化計算を終了し、そのときの解集団の中で最も適応度の高い軌道を誘導軌道として制御へ入力し、誘導システム自体は最適化計算をさらに続けることにより、飛行途中で目標地点へ到達する誘導軌道の生成が行え、最終的に目標地点へ到達することが出来た。

本論文では、1例のみを掲載しているが、様々な飛行環境および機体モデルを用いて本妻生成シミュレーションを行った結果、1回の誘導軌道更新周期で50世代ほどの遺伝的処理を行えば、誘導軌道は、目標地点方向へ飛行する軌道となり、今回の評価項目の中で最も優先度が高い「飛行禁止区域飛行距離」が少ない軌道が優先的に得られるようになっていた。

1回誘導軌道生成で、50世代の遺伝的処理を行う場合、この有翼ロケットの最終目標であるサブオービタル飛行の飛行条件下においては、式(4.5-1)の様な計算が行える。つまり、WIRES#014FPGA実装モデルで使用したFPGAを5枚用いることで、5[s]以内に50世代の遺伝的処理が行えることが解った。

$$\text{計算時間} = 1[\text{s}] \times 5[\text{倍個体数}] \times \frac{50}{200}[\text{倍世代数}] \times \frac{60}{0.6}[\text{高度差}] = 125[\text{s}] (4.5-1)$$

しかし、実際に 50 世代も最適化を行う必要があるかは不明であり、場合によっては FPGA1 枚の計算能力で行える遺伝的処理の世代数で、最終的に目標地点へ到達する飛行結果が得られる可能性がある。

さらに、上昇中に想定される滑空飛行開始位置からの誘導軌道を最適化し続けておくことにより、初回の誘導軌道生成周期から適応度の高い誘導軌道を制御へ入力することが可能であると考ええる。

第5章 結論

5.1. 本研究でわかったこと

本論文では、2種類の手法について提案を行った。

一つ目は、ファジィ集合による階層的適応度計算を応用した DynDGA を用いた軌道生成手法である。ファジィ集合のみを応用した DynDGA による軌道生成手法では、評価指標の正規化に成功したものの、その評価項目の「成長しやすさ」を考慮することが出来ていないことがシミュレーションより確認された。そのため、階層的手法を追加で適応しシミュレーションを行った結果、設計者が設定した評価項目の優先順位に沿って解集団が成長していく様子が見られた。さらに、自動で重み係数を調整するアルゴリズムを適応することにより、設計者は優先順位の設定のみを行うことで、それ以降の最適化は、重み係数を自動でアルゴリズムが調整しながら解探索が行われる可能性を示せたものの、現状では効率よく重み係数の調整が行えてるとは言えず、先制優先因子を固定にした場合のほうが、解の探索能力自体が高いこともわかった。

二つ目は、リアルタイム性を有する DynDGA を用いた誘導軌道生成システムの検討において、飛行中に解探索を継続して行うことを前提に、特定世代で最適化を終了するのではなく、実計算時間で最適化を終了する手法である。GA を用いた誘導軌道生成システムは、FPGA を用いることにより、オンボードリアルタイム生成が可能であることが、カイトプレーンを用いた飛行実証によって確認された。また、DynDGA を用いた誘導軌道生成システムをサブオービタルスケールで使用し、尚且つ飛行中に誘導軌道を更新（再生成）するシミュレーションを行い、結果として、目標地点へ到達する軌道が得られたが、誘導コマンドが不連続であったり、計算時間がカイトプレーンのシミュレーションと比較して大幅に増加していたりすることがわかった。計算時間の増加は、飛行距離の増加によって運動方程式の積分ステップが増加したことが原因であり、この問題を解決するには、計算機の並列数を増やすか、計算手法を変更するしかない。そこで、提案手法を用いることにより、計算時間が長く必要となる飛行距離が長い状況では、最適化にかけられる時間も長いため、飛行途中で誘導軌道を更新することによって目標地点へ到達することができ、飛行距離が短い状況では、必要となる計算時間も短いため、最適化処理が進み、優良な解軌道を初回から使用できることから、ほぼ全ての飛行環境において目標地点へ到達できることがわかった。これを元に、DynDGA のオンボードリアルタイム誘導軌道システムの実現可能性を示せた。

5.2. 今後の課題

DynDGA を用いた誘導軌道生成アルゴリズムに関する課題は、最適化変数で使用している制御入力を表現したフーリエ級数の係数では、遺伝的处理の中で変更した係数の値と、その個体によって得られる解軌道との間に、因果関係が薄く、全ての遺伝子の値がマッチした場合に、最適性が高い軌道が得られることから、非効率的であることがあげられる。しかし、従来用いられているような制御入力の時系列データでは、最適化に時間が掛かってしまう。そこで、ダイナミックインバージョンとベジエ曲線を利用した軌道表現を用い[24]、最適化を行う手法が検討の価値があると考ええる。

係数自動調整手法についても、成長率調整係数の評価が行えていないことから、今後シミュレーションを行い、総合的に本手法の評価を行う必要がある。

また、実飛行を想定したオンボードリアルタイム誘導軌道生成システムに関する課題は、今後実際の飛行実証を行うため、オンボードへ実装し、アルゴリズムが有用であるかの確認をする必要があると考えられる。また、制御入力コマンドが軌道再生成の際に、不連続となってしまうことに関して、フーリエ級数を逆算から求めることは不可能なため、出力するコマンドにフィルターを設け、そのフィルターを通したコマンドで最適化の誘導軌道を算出することで解決できるのではないかと考えられる。

迎角プロファイルにおいて、一部の軌道生成結果に制御入力が振動してしまう結果が見られた。この対策として、機体の応答性と固有振動数などから最適化変数の振動数の項について制限を設ける必要がある。

謝辞

本研究を進めるに当たり、多くの方々にお世話になりました。ここに深く感謝の意を表します。

本研究の遂行に当たり、終始適切かつ熱心なご指導を頂きました九州工業大学 米本 浩一教授、藤川 貴弘助教に謹んで感謝の意を表します。

本論文を纏めるに当たり、有益なご助言と御討論を行っていただいた九州工業大学 黒木 秀一教授、相良 慎一教授、趙 孟佑教授に甚大なる謝意を表します。

海外でご活躍の噂を伺っております松本 剛明助教にも多くのご指導を頂きましたことに深く感謝致します。

本研究に取りかかる際に前任者として懇切丁寧にお教えいただきました宮本信太郎博士、後輩として長く支えてくれた開田 悟史君に深謝を表します。

様々な場面で共に切磋琢磨し、時に励まし合ってきた同輩の GOSSAMSETTI Guna Surendra 君と浦 優介君に、尽きない感謝を表します。

最後に、本研究に関して、多くの助言を頂きました九州工業大学 宇宙システム研究室の先輩方、同輩、後輩、相良研究室の皆様にご心よりお礼申し上げます。

平成 31 年 1 月

市毛 優智

参考文献

- [1] 的川泰宣: 宇宙へのはるかな旅 科学全書 31, 大月書店, 1989
- [2] 田中一郎, NASDA(宇宙開発事業団): 21 世紀の宇宙開発 最先端技術をもたらす宇宙利用の幕開け, ジャーナリスト OB 倶楽部 情報資料センター, 2001
- [3] Zheng Z., Wang Y. and Xie F.: Entry optimization computation using simplex algorithm reference trajectory programming, Proceedings of the Second Symposium International Computer Science and Computational Technology (ISCST '09), pp.120-123, Huangshan, China, 2009.
- [4] Yokoyama Y.: Trajectory optimization of space plane using genetic algorithm combined with gradient method, ICAS 2003, pp. 513.1-513.10, 2002.
- [5] 宮本信太郎, 奥田浩詞, 米本浩一: FPGA を用いた GA リアルタイム最適誘導飛行システム構想, 日本航空宇宙学会 西部支部講演会 2009, pp.227-230, 2009.
- [6] Miyamoto S. and Yonemoto K.: Trajectory Optimization Using Algorithm and Its Implementation on FPGA for Real Time System, 2010 Asia-Pacific International Symposium on Aerospace Technology (APISAT-2010), pp.648-651, Xi'an, China, 2010.
- [7] Miyamoto S., Narumi T., Matsumoto T. and Yonemoto K.: Real-Time Guidance System by Implementing Genetic Algorithm on Field Programmable Gate Array, 28th International Symposium on Space Technology and Science (ISTS), 2011-d-42, Okinawa, 2011.
- [8] Miyamoto S., Narumi T., Matsumoto T. and Yonemoto K.: Real-time guidance of reusable winged space plane using genetic algorithm implemented on field programmable gate array, 2012 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM2012), pp.349-354, Kaohsiung, Taiwan, 2012.
- [9] Miyamoto S., Matsumoto T., Yonemoto K.: Dynamic Distributed Genetic Algorithm Using Hierarchical Clustering for Flight Trajectory Optimization of Winged Rocket, The 12th International Conference on Machine Learning and Applications (ICMLA'13), pp.295-298, Miami, FL, USA, 2013.

- [10]電気学会進化技術応用調査専門委員会: 進化技術ハンドブック第1巻, 近代科学社, 2011.
- [11]廣安知之, 三木光範, 佐野正樹, 谷村勇輔: 2 個体分散遺伝的アルゴリズム, モデル化と問題解決, pp.37-40, 2000.
- [12]三木光範, 廣安知之, 中村康範: 遺伝的アルゴリズムの分散並列化に関する研究 (踏み石モデルによる分散遺伝的アルゴリズム), 日本機械学会論文集 (A 編), 63 巻 638 号, pp.2177-2183, 2000.
- [13]村田裕章, 越野亮, 白山政敏, 林貴宏, 木村春彦: 改善集団と改悪集団による進化的停滞を回避する遺伝的アルゴリズムと時間枠つき配送計画問題への適用, 情報処理学会論文誌: 数理モデル化と応用 Vol.47, No. SIG 14(TOM 15), pp.1-8, 2006.
- [14]Ueno S. and Watanabe R: Optimum Emergency Guidance and Control Law of Future Space Transportation System using Homotopy Method, Proc. of 2008 JSASS-KSAS Joint International Symposium on Aerospace Engineering, pp.500-503, 2008.
- [15]文部省宇宙科学研究所 HIMES 飛翔体小研究会: HIMES 飛翔体 (有翼ロケット実験機) 概念設計, 1987.
- [16]J. H. Holland: Adaptation in Natural and Artificial Systems, University of Michigan Press, 1975.
- [17]佐藤浩, 小野功, 小林重信: 遺伝アルゴリズムにおける世代交代モデルの提案と評価, 人工知能学会誌, Vol.12, No.5, 1997, pp. 734-744, 1997.
- [18]小林重信: 実数値 GA のフロンティア, 人工知能学会誌, Vol.24, No.1, pp.128-143, 2009.
- [19]高木幹雄, 下田陽久: 画像解析ハンドブック, 東京大学出版会, 1993.
- [20]Petr, C., Christian G. F. and Carles N.: Fuzzy Logic, Stanford Encyclopedia of Philosophy, 2017.
- [21]Suzuki, S. and Yoshizawa, T.: Multiobjective Trajectory Optimization by Goal Programming with Fuzzy Decisions, Journal of Guidance, Control, and Dynamics, Vol. 17, No. 2, pp. 297-303, 1994.

- [22]デザインテクノロジー, CQ 出版, 2009.
- [23]Itakura K., Yonemoto K. and Ohki T.: Real-Time Guidance Calculation of Rapid Trajectory Planner for Navigation in Complex Terrain using Random Search and Dynamics Filter, The 2015 Asia-Pacific International Symposium on Aerospace Technology, pp.570-577, 2015
- [24]峰朝雅晴: 遺伝的アルゴリズム その理論と先端的手法, 森北出版, 2008.
- [25]樋口隆英, 筒井茂義, 山村雅幸: 実数値 GA におけるシンプレックス交叉の提案, 人工知能学会誌, Vol.16, No.1, pp.147-155, 2001.
- [26]喜多一, 小野功, 小林重信: 実数値 GA のための正規分布交叉の多数の親を用いた拡張法の提案, 計測自動制御学会論文集, Vol.36, No.10, pp.875-883, 2000.
- [27]三木光範, 廣安知之, 金子美華, 畠中一幸: 環境分散型並列遺伝的アルゴリズム, 電子情報通信学会技術研究報告. AI, 人工知能と知識処理 99(96), pp.87-94, 1999.
- [28]高木幹雄, 下田陽久: 画像解析ハンドブック, 東京大学出版会, 1993.
- [29]小野田崇: サポートベクターマシン, オーム社, 2007.
- [30]T.コホネン: 自己組織化マップ, シュプリンガー・フェアラーク東京, 2002.
- [31]M. R. Anderverg: クラスタ分析とその応用, 内田老鶴圃, 1988.
- [32]デザインテクノロジー, CQ 出版, 2009.
- [33]J. E. Volder: The CORDIC Trigonometric Computing Technique, IRE Transactions on Electronic Computers, vol, EC-8, no.3, pp.330-334, 1959.
- [34]Xilinx <http://japan.xilinx.com/products/silicon-devices/fpga/index.htm>
- [35]I. Jolliffe: Principal Component Analysis, 2nd ed., Springer, 2002.
- [36]SAS Institute Inc. : Cubic Clustering Criterion, SAS Technical Report A-108, 1998.

付録 A 有翼ロケット実験機アビオニクス構成

A1. WIRES#014-3A アビオニクス構成

A2. WIRES#015 アビオニクス構成

付録 B 定常滑空近似

付録 C ファジィ集合を用いた階層的適応度計算手法による DynDGA を用いた
誘導軌道生成シミュレーションプログラム構成